



Operations Guide | PUBLIC

SAP IQ 16.1 SP 05

Document Version: 1.0.0 – 2026-01-08

# SAP IQ Utility Reference

# Content

- 1 SAP IQ Utility Reference. . . . . 7**
- 2 Certificate Creation Utility (createcert). . . . . 8**
- 3 Certificate Viewer Utility (viewcert). . . . . 16**
- 4 CP874toUTF8 Database Administration Utility. . . . . 18**
- 5 dbbackup Database Administration Utility. . . . . 19**
- 6 db\_backupheader Database Administration Utility. . . . . 22**
- 7 dbfhide Database Administration Utility. . . . . 27**
- 7.1 Hiding the Contents of .ini Files Using dbfhide. . . . . 28
- 8 dbinfo Database Administration Utility. . . . . 30**
- 9 dbisql Interactive SQL Utility. . . . . 32**
- 10 dblocate Database Administration Utility. . . . . 33**
- 11 dblog Database Administration Utility. . . . . 35**
- 12 dbping Database Administration Utility. . . . . 38**
- 13 dbstop Database Shutdown Utility. . . . . 41**
- 14 dbtran Database Administration Utility. . . . . 43**
- 15 dbvalid Database Administration Utility. . . . . 47**
- 16 dsedit Server Administration Utility. . . . . 50**
- 16.1 Using dsedit in UNIX. . . . . 52
- 16.2 Using dsedit in Windows. . . . . 53
- 16.3 Troubleshoot dsedit. . . . . 55
- 17 iqbackupcheck Backup Verification Utility. . . . . 57**
- 18 iqdsdp Configuration Utility. . . . . 59**
- 19 (Deprecated) iqdsedit Utility. . . . . 60**
- 20 iqdsn Database Administration Utility. . . . . 61**
- 20.1 ODBC Connection Parameters. . . . . 67

<b>21</b>	<b>iqheader Database Administration Utility</b> .....	<b>72</b>
<b>22</b>	<b>iqinit Database Administration Utility</b> .....	<b>75</b>
<b>23</b>	<b>iqocscfg Configuration Utility</b> .....	<b>82</b>
<b>24</b>	<b>iqsqlpp SQL Preprocessor Utility</b> .....	<b>83</b>
<b>25</b>	<b>iqsunload Local Store Unload Utility</b> .....	<b>85</b>
<b>26</b>	<b>iqunload Schema and Data Export Utility</b> .....	<b>86</b>
<b>27</b>	<b>isql Interactive SQL Utility</b> .....	<b>91</b>
<b>28</b>	<b>start_iq Database Server Startup Utility</b> .....	<b>100</b>
28.1	Starting the start_iq Utility .....	101
28.2	Listing All start_iq Switches .....	101
28.3	start_iq Server Options .....	101
	@data Database Server Option .....	107
	@envvar Database Server Option .....	109
	@filename Database Server Option .....	109
	-al Database Server Option .....	110
	-c Database Server Option .....	111
	-ca Database Server Option .....	111
	-cc Database Server Option .....	113
	-ch Database Server Option .....	114
	-cl Database Server Option .....	115
	-cp Database Server Option .....	116
	-cr Database Server Option .....	117
	-cs Database Server Option .....	118
	-ct Database Server Option .....	119
	-cv Database Server Option .....	120
	-dt Database Server Option .....	121
	-ec Database Server Option .....	123
	-ep Database Server Option .....	127
	-es Database Server Option .....	128
	-fips Database Server Option .....	129
	-ga Database Server Option .....	130
	-gb Database Server Option .....	131
	-gc Database Server Option .....	132
	-gd Database Server Option .....	132
	-ge Database Server Option .....	134
	-gf Database Server Option .....	135
	-gk Database Server Option .....	135

-gl Database Server Option. . . . .	136
-gm Database Server Option. . . . .	137
-gn Database Server Option. . . . .	138
-gp Database Server Option. . . . .	139
-gr Database Server Option. . . . .	139
-gss Database Server Option. . . . .	140
-gt Database Server Option. . . . .	140
-gta Database Server Option. . . . .	141
-gtc Database Server Option. . . . .	143
-gu Database Server Option. . . . .	144
-iqdirectio Database Server Option. . . . .	146
-iqfilereadahead Database Server Option. . . . .	146
-iqfreq Database Server Option. . . . .	147
-iqgovern Database Server Option. . . . .	147
-iqhrt Database Server Option. . . . .	148
-iqlm Database Server Option. . . . .	149
-iqlrt Database Server Option. . . . .	149
-iqmc Database Server Option. . . . .	150
-iqmpx_failover Database Server Option. . . . .	151
-iqmpx_ov Database Server Option. . . . .	151
-iqmpx_reclaimwriterfreelist Database Server Option. . . . .	152
-iqmpx_sn Database Server Option. . . . .	152
-iqmsgnum Database Server Option. . . . .	152
-iqmsgsz Database Server Option. . . . .	153
-iqmt Database Server Option. . . . .	153
-iqnotemp Database Server Option. . . . .	154
-iqnumbercpus Database Server Option. . . . .	155
-iqpartition Database Server Option. . . . .	156
-iqlvmem Database Server Option. . . . .	156
-iqro Database Server Option. . . . .	157
-iqstart Database Server Option. . . . .	157
-iqtc Database Server Option. . . . .	158
-iqtss Database Server Option. . . . .	159
-iqwmem Database Server Option. . . . .	159
-k Database Server Option. . . . .	160
-kl Database Server Option. . . . .	161
-kr Database Server Option (Deprecated). . . . .	161
-krb Database Server Option. . . . .	162
-ks Database Server Option. . . . .	163
-ksc Database Server Option. . . . .	164
-ksd Database Server Option. . . . .	164

-lt Database Server Option. . . . .	165
-m Database Server Option. . . . .	166
-mp Database Server Option. . . . .	167
-n Database Server Option. . . . .	168
-o Database Server Option. . . . .	169
-oe Database Server Option. . . . .	170
-on Database Server Option. . . . .	170
-os Database Server Option. . . . .	171
-ot Database Server Option. . . . .	172
-p Database Server Option. . . . .	173
-pc Database Server Option. . . . .	173
-phl Database Server Option. . . . .	174
-phs Database Server Option. . . . .	175
-pt Database Server Option. . . . .	176
-qi Database Server Option. . . . .	177
-qp Database Server Option. . . . .	177
-qs Database Server Option. . . . .	178
-qw Database Server Option. . . . .	178
-s Database Server Option. . . . .	179
-sb Database Server Option. . . . .	179
-sf Database Server Option. . . . .	180
-sk Database Server Option. . . . .	186
-su Database Server Option. . . . .	188
-tdsl Database Server Option. . . . .	189
-ti Database Server Option. . . . .	190
-tl Database Server Option. . . . .	191
-tmf Database Server Option. . . . .	192
-tq Database Server Option. . . . .	192
-u Database Server Option. . . . .	193
-ud Database Server Option. . . . .	193
-uf Database Server Option. . . . .	194
-ufd Database Server Option. . . . .	194
-ut Database Server Option. . . . .	196
-v Database Server Option. . . . .	196
-x Database Server Option. . . . .	196
-xd Database Server Option. . . . .	198
-xs Database Server Option. . . . .	198
-z Database Server Option. . . . .	201
-ze Database Server Option. . . . .	201
-zl Database Server Option. . . . .	202
-zn Database Server Option. . . . .	203

	-zo Database Server Option. . . . .	204
	-zoc Database Server Option. . . . .	205
	-zr Database Server Option. . . . .	205
	-zs Database Server Option. . . . .	207
28.4	start_iq Database File Parameters. . . . .	208
28.5	start_iq Database Options. . . . .	209
	-a Database Option. . . . .	210
	-ad Database Option. . . . .	211
	-al Database Option. . . . .	212
	-ar Database Option. . . . .	213
	-as Database Option. . . . .	214
	-dh Database Option. . . . .	215
	-ds Database Option. . . . .	215
	-ek Database Option. . . . .	216
	-iqfreq Database Option. . . . .	216
	-m Database Option. . . . .	217
	-n Database Option. . . . .	218
	-r Database Option. . . . .	218
	-sm Database Option. . . . .	219
	-sn Database Option. . . . .	219
	-xp Database Option. . . . .	219
<b>29</b>	<b>stop_iq Database Shutdown Utility. . . . .</b>	<b>221</b>
29.1	When to Stop and Restart the Server. . . . .	221
29.2	Ways to Stop Database Servers. . . . .	222
29.3	Permissions Required to Stop the Server. . . . .	224
29.4	Stopping the Database Server with stop_iq. . . . .	224
	stop_iq Options. . . . .	225
	Stopping Servers in a cron or at Job. . . . .	226
	Servers with Long Paths. . . . .	227
<b>30</b>	<b>Appendix: dbisqlc Interactive SQL Classic Utility (Deprecated). . . . .</b>	<b>228</b>
30.1	Interactive SQL Classic Syntax. . . . .	228
	Interactive SQL Classic Options. . . . .	229
30.2	Function and Special Keys (UNIX). . . . .	229
30.3	Function and Special Keys (Windows). . . . .	230
30.4	Command Recall Keys (Windows). . . . .	231

# 1 SAP IQ Utility Reference

Utility programs are commands that you invoke directly from the operating system. This guide provides information on utility syntax, parameters, and options. Other books in the SAP IQ documentation set provide more context on how to carry out particular tasks.

## 2 Certificate Creation Utility (createcert)

Creates X.509 certificates.

### ⌘ Syntax

```
createcert [ -r | -s ]
```

Option	Prompt name	Description
@<data>	N/A	<p>Reads options from the specified environment variable or configuration file.</p> <p>To protect information in the configuration file, use the File Hiding utility (dbfhide) to encode the contents of the configuration file.</p>
-b <value>	Enter RSA key length (512-16384)	<p>Specifies the RSA key length. The length can be between 512 bits and 16384 bits.</p> <p>This option is ignored when the -s option is specified.</p>
-c <filename>	Enter file path of signer's certificate	<p>Specifies a location and file name for the signer's certificate.</p> <p>If you supply this information, then the generated certificate is a signed certificate. If you do not supply this information, then the generated certificate is a self-signed root certificate.</p> <p>This option is ignored when the -r option is specified.</p>
-ca {<0   1>}	Certificate Authority (Y/N) [N]	<p>Creates a certificate authority certificate that can be used to sign other certificates. By default, certificates are not certificate authorities (0).</p> <p>This option is ignored when the -r option is specified.</p>

Option	Prompt name	Description
<code>-ck &lt;filename&gt;</code>	<b>Enter file path of signer's private key</b>	A location and file name to save the private key associated with the certificate request. This option prompt only appears if you specify a file name for the <code>-c</code> option.
<code>-co &lt;filename&gt;</code>	<b>Enter file path to save certificate</b>	Writes the public certificate to the specified file.  Specify a location and file name to save the certificate; otherwise, the certificate is not saved.  This option is ignored when the <code>-r</code> option is specified.
<code>-cp &lt;password&gt;</code>	<b>Enter password for signer's private key</b>	Specifies the password that was used to encrypt the signer's private key.  Only specify this password if the private key was encrypted.  This option is ignored when the <code>-r</code> option is specified.
<code>-io &lt;filename&gt;</code>	<b>Enter file path to save identity</b>	Specifies a location and file name to save the generated identity.  This option is ignored when the <code>-r</code> option is specified.
<code>-ko &lt;filename&gt;</code>	<b>Enter file path to save private key</b>	Specifies a location and file name to save the private key.
<code>-kp &lt;password&gt;</code>	<b>Enter password to protect private key</b>	Specifies a password with which to encrypt the private key.  If you do not specify a password, the private key is not encrypted. This option only applies if you specify a file name with the <code>-ko</code> option.

Option	Prompt name	Description
<code>-m &lt;value&gt;</code>	Serial number [generate GUID]	<p>Specifies a serial number. The serial number must be a hexadecimal string of 40 digits or less. This number must be unique among all certificates signed by the current signer. If you do not specify a serial number, creatcert generates a GUID as the serial number.</p> <p>This option is ignored when the <code>-r</code> option is specified.</p>
<code>-p1</code>	N/A	Uses PKCS #1 to encode unencrypted RSA private keys instead of the default, PKCS #8.
<code>-r</code>	N/A	<p>Generates a PKCS #10 certificate request.</p> <p>When this option is specified, the following options do not apply:</p> <ul style="list-style-type: none"> <li>• <code>-c</code></li> <li>• <code>-ca</code></li> <li>• <code>-ck</code></li> <li>• <code>-co</code></li> <li>• <code>-cp</code></li> <li>• <code>-m</code></li> <li>• <code>-io</code></li> <li>• <code>-u</code></li> <li>• <code>-x</code></li> <li>• <code>-v</code></li> </ul>
<code>-r0 &lt;filename&gt;</code>	Enter file path to save request	<p>Specifies a location and file name with which to save the PKCS #10 certificate request.</p> <p>This option only applies if you specify the <code>-r</code> option.</p>

Option	Prompt name	Description
<code>-s &lt;filename&gt;</code>	N/A	<p>Specifies the location and file name of the PKCS10 certificate request that is to be signed. The certificate request can be DER or PEM encoded. When this option is specified, the following options do not apply:</p> <ul style="list-style-type: none"> <li>• -b</li> <li>• -ec</li> <li>• -sc</li> <li>• -scn</li> <li>• -sl</li> <li>• -so</li> <li>• -sou</li> <li>• -sst</li> <li>• -t</li> <li>• -x</li> </ul>
<code>-sa &lt;value&gt;</code>	N/A	<p>Specifies which of the following signature algorithms to use: sha1, sha256, sha384, or sha512.</p> <p>The default algorithm is sha256.</p>
<code>-sc &lt;value&gt;</code>	<b>Country Code</b>	<p>Specifies the subject country code.</p> <p>This option is ignored when the -s option is specified.</p>
<code>-scn &lt;value&gt;</code>	<b>Common Name</b>	<p>Specifies the subject common name.</p> <p>This value must be the computer host name on which the certificate is to be used. It should match the Host=host-name that the client would use.</p> <p>This option is ignored when the -s option is specified.</p>
<code>-sl &lt;value&gt;</code>	<b>Locality</b>	<p>Specifies the subject locality.</p> <p>This option is ignored when the -s option is specified.</p>
<code>-so &lt;value&gt;</code>	<b>Organization</b>	<p>Specifies the subject organization.</p> <p>This option is ignored when the -s option is specified.</p>

Option	Prompt name	Description
<code>-sou &lt;value&gt;</code>	<b>Organizational Unit</b>	<p>Specifies the subject organizational unit.</p> <p>This option is ignored when the <code>-s</code> option is specified.</p>
<code>-sst &lt;value&gt;</code>	<b>State/Province</b>	<p>Specifies the subject state or province.</p> <p>This option is ignored when the <code>-s</code> option is specified.</p>
<code>-t &lt;type&gt;</code>	N/A	<p>Specifies the encryption type. Specify RSA.</p> <p>This option is ignored when the <code>-s</code> option is specified.</p>
<code>-u &lt;value, ...&gt;</code>	<b>Key usage</b>	<p>Specifies how the certificate will be used.</p> <p>Specify a comma-separated list of numbers that indicate how the certificate's private key can be used. This option is an advanced option; the default should be acceptable for most situations. The default depends on whether the certificate is a certificate authority or not. The default for a certificate with certificate authority is <a href="#">6,7</a>. The default for a certificate without certificate authority is <a href="#">3,4,5</a>.</p> <p>This option is ignored when the <code>-r</code> option is specified.</p> <p>Values include:</p> <ul style="list-style-type: none"> <li>• 1. Digital Signature</li> <li>• 2. Nonrepudiation</li> <li>• 3. Key Encipherment</li> <li>• 4. Data Encipherment</li> <li>• 5. Key Agreement</li> <li>• 6. Certificate Signing</li> <li>• 7. CRL Signing</li> <li>• 8. Encipher Only</li> <li>• 9. Decipher Only</li> </ul>

Option	Prompt name	Description
<code>-v &lt;years&gt;</code>	<b>Certificate valid for how many years (1-100)</b>	Specifies the number of years (between 1 and 100) that the certificate is valid. After this period, the certificate expires, along with all certificates it signs.  This option is ignored when the <code>-r</code> option is specified.
<code>-x</code>	N/A	Generates a self-signed certificate.  This option is ignored when the <code>-r</code> option is specified.
<code>-3des</code>	N/A	Encrypts the private key using 3DES encryption. Use this option to create a server identity file that can be used by older database servers running the Certicom encryption module as well as database servers that use the OpenSSL encryption module. Do not use this option if you are running in FIPS mode.

## Privileges

None.

## Remarks

Users typically go to a third party to purchase certificates. These certificate authorities provide their own tools for creating certificates. The following tools may be especially useful to create certificates for development and testing purposes, and can also be used for production certificates.

To create a signed certificate, you can specify no options and be prompted by `createcert` for the required values. Or, you can specify the required options for your certificate.

To break up the process into two steps, for example so one person creates a request and another person signs it, the first person can run `createcert` with `-r` to create a request and the second person can sign the request by running `createcert` with `-s`.

To create a root certificate (a certificate that signs other certificates), create a self-signed certificate with certificate authority and use the default signing options. For example, specify `y` for the `-ca` option and specify `6,7` for the `-u` option.

Private keys encrypted using that module were encrypted using 3DES encryption, instead of AES.

SQL Anywhere supports PKCS #1, PKCS #8 (default) and PKCS #10. If you use another format such as PKCS #12, convert the certificate into a supported format with an OpenSSL utility.

## Example

The following example creates a self-signed certificate with the following characteristics:

- 1024-bit RSA key
- Generated serial number
- Valid for 5 years
- Not a certificate authority
- Used for digital signature, key encipherment, data encipherment, and key agreement
- Private key with the password sqlkey

```
createcert -t rsa -b 1024 -sc CA -sst ON -sl Waterloo -so MyCompany -sou MyUnit  
-scn test -x -m 0 -v 5 -ca 0 -u 1,3,4,5 -co root-cert.pem -ko root-key.pem -io  
root-id.pem -kp sqlkey
```

The following example creates a certificate signed by ca-key with the following characteristics:

- 1024-bit RSA key
- Generated serial number
- Valid for 5 years
- Not certificate authority
- Used for digital signature, key encipherment, data encipherment, and key agreement.
- Private key with the password sqlkey:

```
createcert -t rsa -b 1024 -sc CA -sst ON -sl Waterloo -so MyCompany -sou MyUnit  
-scn test -c ca-cert.pem -ck ca-key.pem -cp cakeypass -m 0 -v 5 -ca 0 -u 1,3,4,5  
-co root-cert.pem -ko root-key.pem -io root-id.pem -kp sqlkey
```

The following example creates a signed certificate. In the example, because no file name is provided for the signer's certificate, the certificate is a self-signed root certificate.

```
createcert  
SQL Anywhere X.509 Certificate Generator Version 17.1.0.1691  
Warning: The certificate will not be compatible with older versions  
of the software including version 12.0.1 prior to build 3994 and version 16.0  
prior to build 1691. Use the -3des switch if you require compatibility.  
Enter RSA key length (512-16384): 1024  
Generating key pair...  
Country Code: CA  
State/Province: Ontario  
Locality: Waterloo  
Organization: MyCompany  
Organizational Unit: Engineering  
Common Name: Test Certificate  
Enter file path of signer's certificate:  
Certificate will be a self-signed root  
Serial number [generate GUID]:  
Generated serial number: 3ccd9e6b0c8c45d98f45f8a3f51d6ba0  
Certificate valid for how many years (1-100): 10
```

```
Certificate Authority (Y/N) [N]: n
1. Digital Signature
2. Nonrepudiation
3. Key Encipherment
4. Data Encipherment
5. Key Agreement
6. Certificate Signing
7. CRL Signing
8. Encipher Only
9. Decipher Only
Key Usage [1,3,4,5]: 1,3,4,5
Enter file path to save certificate: cert.pem
Enter file path to save private key: key.pem
Enter password to protect private key: pwd
Enter file path to save identity: id.pem
```

The following example creates a root certificate (a certificate that signs other certificates) that is a self-signed root certificate with Certificate Authority that uses the default signing. The response to the **Certificate Authority** prompt or the `-ca` option should be **y** and response to the Key usage prompt or the `-u` option should be **6,7** (the default).

```
Certificate Authority (Y/N) [N]: y
1. Digital Signature
2. Nonrepudiation
3. Key Encipherment
4. Data Encipherment
5. Key Agreement
6. Certificate Signing
7. CRL Signing
8. Encipher Only
9. Decipher Only
Key Usage [6,7]: 6,7
```

## 3 Certificate Viewer Utility (viewcert)

Displays values within a Public Key Infrastructure (PKI) object, converts the encoding of PKI objects, or encrypts and decrypts private keys.

### ⌘ Syntax

```
viewcert [ <options> ] <input-file>
```

Option	Description
@<data>	Reads options from the specified environment variable or configuration file.  To protect information in the configuration file, use the File Hiding utility (dbfhide) to encode the contents of the configuration file.
-d	DER-encodes the output. This option is only useful with the -o option. It cannot be used with -p. By default, viewcert outputs the PKI object in a readable text format.
-ip <input-password>	Specifies the password needed to decrypt the private key if the <input-file> contains an encrypted private key.
-o <filename>	Specifies the file that viewcert should write the output to. By default, viewcert writes the output to the command prompt window where it is running.
-op <output-password>	Specifies the password viewcert should use to encrypt a private key. This option is only useful with -d or -p. By default, private keys are not encrypted.
-p	PEM-encodes the output. This option is only useful with the -o option. It cannot be used with -d. By default, viewcert outputs the PKI object in a readable text format.
-p1	Uses PKCS #1 to encode unencrypted RSA private keys instead of the default, PKCS #8.
-q	Runs in quiet mode--messages are not displayed.
<input-file>	Specifies a file that must be a DER- or PEM-encoded PKI object.
-3des	Specifies to use 3DES encryption instead of AES when specifying -p, -ip, and -op options. Do not use this option if you are running in FIPS mode.

## Privileges

None.

## Remarks

The viewcert utility can be used to view the following types of PKI objects:

- X.509 certificates
- certificate requests
- private keys
- certificate revocation lists (CRLs)

Viewcert can also be used to convert between DER and PEM encoding types and to encrypt or decrypt private keys.

The viewcert utility supports RSA objects.

## Example

The following example allows you to view the sample RSA certificate that is included with SAP IQ. From the `%IQDIRSAMP%\Certificates` directory, run the following command:

```
viewcert rsaroot.crt
```

This example produces the following output:

```
SQL Anywhere X.509 Certificate Viewer Version 17.0.0.1245
Common Name: Sample RSA Root Certificate
Country Code: CA
State/Province: ON
Locality: Waterloo
Organization: SAP
Organizational Unit: SQL Anywhere
Issuer: Sample RSA Root Certificate
Serial Number: 101
Issued: Apr 19, 2015 13:49:00
Expires: Apr 20, 2025 13:49:00
Signature Algorithm: RSA, SHA256
Key Type: RSA
Key Size: 2048 bits
Basic Constraints: Is a certificate authority, path length limit: 10
Key Usage: Certificate Signing, CRL Signing
```

## 4 CP874toUTF8 Database Administration Utility

The `CP874toUTF8` utility converts data in the CP874 character set into the UTF8 collation that is supported by SAP IQ for the Thai language.

### ↔ Syntax

```
CP874toUTF8 [ <CP874InputFile> ]
```

### Remarks

You can also use this utility to load data in the CP874 character set without converting it to UTF8.

You can run this utility only from the command prompt.

`CP874toUTF8` reads the named file in the CP874 character set (or standard input if no files are named) and prints the UTF8 conversion to standard output.

### ⓘ Note

Files with embedded NULL characters ('\0') are not converted correctly. Remove such characters before running this utility.

Use caution with large data files; the UTF8 output can be up to three times larger than the input data. Input and output file size must both be within operating system limits.

`CP874toUTF8` returns a 0 exit status upon successful completion. A nonzero exit status indicates that an error occurred. The `CP874toUTF8` utility writes all error messages to `stderr`.

`CP874toUTF8` calls the International Components for Unicode (ICU) library to perform the data conversion. If ICU cannot convert the data, `CP874toUTF8` reports a conversion error. If the conversion fails, divide the file into smaller sections for conversion, to isolate the point of failure. If you cannot resolve the problem, contact Technical Support.

# 5 dbbackup Database Administration Utility

The dbbackup utility makes a copy of the transaction log of a running SAP IQ database.

## Note

To back up an entire SAP IQ database, always use `BACKUP DATABASE` instead of `dbbackup`. `BACKUP DATABASE` backs up all database files, and is the only way to back up the catalog store.

To truncate the transaction log on a simplex or multiplex server, use the `-m` option in your configuration file or `start_iq` command. The server may be running or stopped.

## Syntax

```
dbbackup [ <options> ] <target-directory>
```

## Parameters

### options

#### @data

Reads options from the specified environment variable or configuration file. If both exist, and share the same name, SAP IQ uses the environment variable.

To protect passwords or other information in the configuration file, use the File Hiding utility (`dbfhide`) to obfuscate configuration file contents.

#### -c "keyword=value; ... "

Supplies database connection parameters. If you do not specify the connection parameters, connection parameters from the `SQL CONNECT` environment variable are used, if set. The user ID must have the `BACKUP DATABASE` system privilege.

#### -l file

(Lowercase L) Sends a live backup of the transaction log to a file. Enables a secondary system to be brought up rapidly in the event of server failure. A live backup does not terminate, but continues while the server runs. It runs until the primary server becomes unavailable. At that point, it shuts down, but the backed up log file is intact and can be used to quickly bring up a secondary system.

The live backup of the transaction log is always the same length or shorter than the active transaction log. When a live backup is running and another backup restarts the transaction log (`dbbackup -x`), the live backup automatically truncates the live backup log and restarts the live backup at the beginning of the new transaction log.

#### -o

Writes output messages to the named file.

**-q**

Quiet mode; does not display output messages. This option is available only when you run this utility from a command prompt.

**-r**

Copies the old transaction log to a new name and starts a new empty log. The following steps occur:

- SAP IQ copies and saves the current working transaction log to the directory specified in the command.
- SAP IQ keeps the current transaction log in its current directory, but renames it using the format `yyymmdd<xx>.log`, where `<xx>` are sequential characters starting at `<AA>` and running to `<ZZ>`, and `<yyymmdd>` represents the current year, month, and day. This file is then no longer the current transaction log.
- SAP IQ generates a new transaction log file that contains no transactions. The new file has the name of the former current transaction log and becomes the current transaction log.

**-t**

Backs up only the transaction log file. This can be used as an incremental backup since the transaction log can be applied to the most recently backed up copy of the database files.

**-xo filename**

Truncate (delete and restart) the transaction log

**target-directory**

The directory to which the backup files are copied. If the directory does not exist, SAP IQ creates it. The parent directory must exist.

## Privileges

Requires the BACKUP DATABASE and SERVER OPERATOR system privileges.

## Remarks

The `dbbackup` utility allows you to back up the transaction log while other applications or users are using the database. Backup file names are the same as the database file names.

The `dbbackup` utility truncates the database name to 70 characters and creates a target file with a truncated name. SAP IQ uses `dbbackup` when synchronizing secondary servers. Due to `dbbackup` restrictions, database names must be less than 70 characters long.

If you have adequate disk space, use `-r` to preserve the existing log file under a new name and start a new empty log. If disk space is limited, use `-xo` instead to truncate the existing log.

Exit codes are 0 (success) or nonzero (failure).

SERVER is the default logical server for `dbbackup`.

See *BACKUP DATABASE Statement* in *SAP IQ SQL Reference*.

## Examples

The following Windows command backs up the transaction log from the `iqdemo` database running on the `sample_server` server into the directory `iqbackup`, connecting as user ID `DBA` with password `sql`:

```
dbbackup -c "eng=sample_server;dbn=iqdemo;  
uid=DBA;pwd=<password>" c:\sample\iqbackup
```

## Related Information

[dbfhide Database Administration Utility \[page 27\]](#)

[-m Database Server Option \[page 166\]](#)



```

Redo log Branch offset: "0"
Last commit time: "2015-05-18 13:57:12.429+00:00"
Backup size: "25408 kb"
Current timeline GUID: "403a4e18-70fa-11e4-8000-dad0599922e0"
Current timeline creation time: "2014-11-21 07:14:44.965919+00:00"
Previous timeline GUID: "00000000-0000-0000-0000-000000000000"
Previous timeline creation time: "0000-01-00 00:00:00.000000+00:00"
Current log GUID: "9c4d595c-fd22-11e4-8000-cd8967e5e385"
Current log creation time: "2015-05-18 13:56:22.098168+00:00"
Previous log GUID: "404dd618-70fa-11e4-8000-dad0599922e0"
DB Header Record:
DB Name: "asiq.db"
IQ Block Size: "8192"
IQ Page Size: "131072"
Number of IQ Blocks: "2748"
IQ Root Block: "2610"
SA Block Size: "4096"
Number of SA Blocks: "856"
Number of Dbspaces: "7"
IQ Block Factor: "25"
Number of dbspace definitions: 7
DB Space Record:
DBSpace Name: "system"
DBSpace ID: "0"
Filler1: 0x
Alter ID: "0"
Create ID: "0"
DBSpace Type: "1"
Segment Type: "0"
Number of DBFiles: "1"
Is Readonly: "no"
Is Online: "yes"
Is Backed Up: "yes"
Filler2: 0x0
Number of dbfile definitions in dbspace "system": 1
  DB File Record:
    File Name: "system"
    File Path: "/machine1/users/user1/user1_lint13d/asiq.db"
    File ID: "0"
    DBSpace ID: "0"
    Alter ID: "0"
    Create ID: "0"
    Start Block: "0"
    Number of Blocks: "856"
    Number of Blocks Backed Up: "856"
    Data Offset: "0"
    Is Readonly: "no"
    Is Backed Up: "yes"
    Is Backed Up Distributed: "no"
    Filler: 0x0
    Filler: 0x0
DB Space Record:
DBSpace Name: "IQ_SYSTEM_LOG"
DBSpace ID: "32702"
Filler1: 0x
Alter ID: "0"
Create ID: "0"
DBSpace Type: "2"
Segment Type: "6"
Number of DBFiles: "1"
Is Readonly: "no"
Is Online: "yes"
Is Backed Up: "no"
Filler2: 0x0
Number of dbfile definitions in dbspace "IQ_SYSTEM_LOG": 1
  DB File Record:
    File Name: "IQ_SYSTEM_LOG"
    File Path: ""

```

```

File ID: "32702"
DBSpace ID: "32702"
Alter ID: "0"
Create ID: "0"
Start Block: "0"
Number of Blocks: "0"
Number of Blocks Backed Up: "0"
Data Offset: "0"
Is Readonly: "no"
Is Backed Up: "no"
Is Backed Up Distributed: "no"
Filler: 0x0
Filler: 0x0
DB Space Record:
DBSpace Name: "IQ_SYSTEM_MAIN"
DBSpace ID: "16384"
Filler1: 0x
Alter ID: "0"
Create ID: "0"
DBSpace Type: "2"
Segment Type: "1"
Number of DBFiles: "1"
Is Readonly: "no"
Is Online: "yes"
Is Backed Up: "yes"
Filler2: 0x0
Number of dbfile definitions in dbspace "IQ_SYSTEM_MAIN": 1
  DB File Record:
  File Name: "IQ_SYSTEM_MAIN"
  File Path: "asiq.iq"
  File ID: "16384"
  DBSpace ID: "16384"
  Alter ID: "0"
  Create ID: "0"
  Start Block: "1"
  Number of Blocks: "12800"
  Number of Blocks Backed Up: "2748"
  Data Offset: "65536"
  Is Readonly: "no"
  Is Backed Up: "yes"
  Is Backed Up Distributed: "no"
  Filler: 0x0
  Filler: 0x0
DB Space Record:
DBSpace Name: "IQ_SYSTEM_TEMP"
DBSpace ID: "16385"
Filler1: 0x
Alter ID: "0"
Create ID: "0"
DBSpace Type: "2"
Segment Type: "3"
Number of DBFiles: "1"
Is Readonly: "no"
Is Online: "yes"
Is Backed Up: "no"
Filler2: 0x0
Number of dbfile definitions in dbspace "IQ_SYSTEM_TEMP": 1
  DB File Record:
  File Name: "IQ_SYSTEM_TEMP"
  File Path: "asiq.iqtmp"
  File ID: "16385"
  DBSpace ID: "16385"
  Alter ID: "0"
  Create ID: "0"
  Start Block: "1"
  Number of Blocks: "12800"
  Number of Blocks Backed Up: "0"
  Data Offset: "65536"

```

```

    Is Readonly: "no"
    Is Backed Up: "no"
    Is Backed Up Distributed: "no"
    Filler: 0x0
    Filler: 0x0
DB Space Record:
DBSpace Name: "IQ_SYSTEM_MSG"
DBSpace ID: "16386"
Filler1: 0x
Alter ID: "0"
Create ID: "0"
DBSpace Type: "2"
Segment Type: "2"
Number of DBFiles: "1"
Is Readonly: "no"
Is Online: "yes"
Is Backed Up: "no"
Filler2: 0x0
Number of dbfile definitions in dbspace "IQ_SYSTEM_MSG": 1
  DB File Record:
    File Name: "IQ_SYSTEM_MSG"
    File Path: "asiq.iqmsg"
    File ID: "16386"
    DBSpace ID: "16386"
    Alter ID: "0"
    Create ID: "0"
    Start Block: "0"
    Number of Blocks: "0"
    Number of Blocks Backed Up: "0"
    Data Offset: "0"
    Is Readonly: "no"
    Is Backed Up: "no"
    Is Backed Up Distributed: "no"
    Filler: 0x0
    Filler: 0x0
DB Space Record:
DBSpace Name: "IQ_SHARED_TEMP"
DBSpace ID: "32703"
Filler1: 0x
Alter ID: "0"
Create ID: "0"
DBSpace Type: "2"
Segment Type: "0"
Number of DBFiles: "0"
Is Readonly: "no"
Is Online: "yes"
Is Backed Up: "no"
Filler2: 0x0
Number of dbfile definitions in dbspace "IQ_SHARED_TEMP": 0
  DB Space Record:
    DBSpace Name: "translog"
    DBSpace ID: "13"
    Filler1: 0x
    Alter ID: "0"
    Create ID: "0"
    DBSpace Type: "1"
    Segment Type: "0"
    Number of DBFiles: "1"
    Is Readonly: "no"
    Is Online: "yes"
    Is Backed Up: "no"
    Filler2: 0x0
  Number of dbfile definitions in dbspace "translog": 1
    DB File Record:
      File Name: "translog"
      File Path: "/machine1/users/user1/user1_lint13d/asiq.log"
      File ID: "13"
      DBSpace ID: "13"

```

```
Alter ID: "0"  
Create ID: "0"  
Start Block: "0"  
Number of Blocks: "2"  
Number of Blocks Backed Up: "0"  
Data Offset: "0"  
Is Readonly: "no"  
Is Backed Up: "no"  
Is Backed Up Distributed: "no"  
Filler: 0x0  
Filler: 0x0  
Store Boundary Record:
```

# 7 dbfhide Database Administration Utility

The `dbfhide` utility is a file hiding utility you use to add simple encryption to configuration files and initialization files to hide the contents of each file.

## ⌘ Syntax

```
dbfhide <original-configuration-file> <encrypted-configuration-file>
```

## Parameters

### **original-configuration-file**

Name of the original file.

### **encrypted-configuration-file**

Name for the new obfuscated file.

## Privileges

None.

## Remarks

Configuration files are used by some utilities to hold command-line options. These options can contain a password. You can use the `dbfhide` utility to add simple encryption to configuration files and `.ini` files used by SAP IQ and its utilities, and thereby obfuscate the contents of the file. The original file is not modified. Once you add simple encryption to a file, there is no way to remove it. To change an obfuscated file, keep a copy of the original file that you can modify and obfuscate again.

## Examples

Create a configuration file that starts the personal database server and the sample database. The file should set a cache of 10MB, and name this instance of the personal server "Elora". The configuration file would be written as follows:

```
# Configuration file for server Elora -n Elora -c 10M path\iqdemo.db
```

Lines beginning with # are treated as comments.

Name the file `sample.txt`. To start the database using this configuration file, enter:

```
start_iq @sample.txt
```

Add simple encryption to the configuration:

```
dbfhide sample.txt encrypted_sample.txt
```

Use the `encrypted_sample.txt` file to start a database:

```
start_iq @encrypted_sample.txt
```

For more information about using configuration files, see *Using Configuration Files* in the *SAP IQ Installation and Update Guide* for your platform.

The following command adds simple encryption to the `asaldap.ini` file:

```
dbfhide asaldap.ini encrypted_asaldap.ini
```

#### In this section:

[Hiding the Contents of .ini Files Using dbfhide \[page 28\]](#)

Obfuscate an .ini file using the File Hiding (dbfhide) utility.

## Related Information

[dbbackup Database Administration Utility \[page 19\]](#)

[-m Database Server Option \[page 166\]](#)

## 7.1 Hiding the Contents of .ini Files Using dbfhide

Obfuscate an .ini file using the File Hiding (dbfhide) utility.

### Procedure

1. Save the file with a different name.

```
rename saldap.ini saldap.ini.org
```

If you do not keep a copy of the original file, then you cannot modify the contents of the file once it has been obfuscated.

2. Obfuscate the file with the File Hiding utility, giving the obfuscated file the required file name:

```
dbfhide saldap.ini.org saldap.ini
```

3. Protect the `saldap.ini.org` file using file system or operating system protection, or store the file in a secure location.

To change the `saldap.ini` file, edit the `saldap.ini.org` file and repeat step 2.

#### Caution

(UNIX) You should not add simple encryption to the `.odbc.ini` system information file with the `dbfhide` File Hiding utility on UNIX unless you will only be using SAP IQ data sources. If you plan to use other data sources, then obfuscating the contents of the `.odbc.ini` file may prevent other drivers from functioning properly.

`dbfhide` does not accept the `@data` parameter to read in options from a configuration file.

## 8 dbinfo Database Administration Utility

The `dbinfo` utility displays information about a database catalog store. The information returned by `dbinfo` does not reflect the IQ main store.

Any valid user ID can run `dbinfo`, but to obtain page usage statistics you need the `MANAGE ANY DBSPACE` system privilege.

### ↳ Syntax

```
dbinfo [ <options> ]
```

### Parameters

#### options

**-c "keyword=value; ..."**

Specify connection parameters. See *Connection and Communication Parameters Reference* in *SAP IQ Administration: Database*.

**-o filename**

Write output messages to the named file.

**-q**

Operate quietly; do not display output messages.

**-u**

Output page usage statistics. Display information about the usage and size of all catalog store tables, including system and user-defined tables. You can only request page usage statistics if no other users are connected to the database.

### Privileges

Any user can run `dbinfo`. However, to obtain page usage statistics, you must have the `MANAGE ANY DBSPACE` system privilege.

## Remarks

The `dbinfo` utility indicates when the database was created, the name of any transaction log file or log mirror that is maintained, the catalog store page size, the version of installed Java classes, and other information. Optionally, it can also provide catalog table usage statistics and details.

# 9 dbisql Interactive SQL Utility

See *SAP IQ Interactive SQL Guide* in the SAP IQ documentation.

## Related Information

[SAP IQ Interactive SQL Guide](#)

# 10 dblocate Database Administration Utility

The `dblocate` utility is a server location utility that assists in diagnosing connection problems by locating databases on the immediate TCP/IP network.

## ⌘ Syntax

```
dblocate [ <options> ]
```

## Parameters

### options

#### **-d**

Lists the server name and address, for each server found, followed by a comma-separated list of databases running on that server. If the list exceeds 160 characters, it is truncated and ends with an ellipsis (...).

#### **-dn database-name**

Lists the server name and address, for servers running a database with the specified name. If the list exceeds 160 characters, it is truncated and ends with an ellipsis (...).

#### **-dv**

Displays the server name and address, for each server found, listing each database running on that server on a separate line. The list is not truncated, so this option can be used to reveal lists that are truncated when the `v` option is used.

#### **-n**

Lists IP addresses in the output, rather than computer names. This may improve performance since looking up computer names may be slow.

#### **-o filename**

Writes output messages to the named file.

#### **-p portnumber**

Displays the server name and address only for servers using the specified TCP/IP port number. The TCP/IP port number must be between 1 and 65535.

#### **-q**

Runs in quiet mode; messages are not displayed.

#### **-s name**

Displays the server name and address only for servers with the specified server name. If this option is used, the `-ss` option should not be used (if both options are used, it is likely that no matching servers will be found).

#### **-ss substr**

Displays the server name and address only for servers that contain the specified substring anywhere in the server name. If this option is used, the `-s` option should not be used (if both options are used, it is likely that no matching servers will be found).

**-v**

Displays the full server name. By default, `dblocate` truncates database server names that are longer than 40 bytes.

## Privileges

None.

## Remarks

`dblocate` locates any SAP SQL Anywhere or SAP IQ database servers running over TCP/IP on the immediate network. It prints a list of database servers and their addresses.

Depending on your network, the utility may take several seconds before printing its results.

## Related Information

[start\\_iq Database Options \[page 209\]](#)

# 11 dblog Database Administration Utility

The `dblog` utility is a transaction log utility that displays or changes the name of the transaction log or transaction log mirror associated with your database.

You can also use `dblog` to stop a database from maintaining a transaction log mirror, or start maintaining a transaction log mirror.

SAP IQ automatically handles the creation and deletion of the transaction log for a database. The database must run with a transaction log. The server will not start without a transaction log. A transaction log mirror is a duplicate copy of a transaction log, maintained in tandem by the database.

## Note

Although not required, use a transaction log mirror, especially if you do not frequently back up your SAP IQ database.

## Syntax

```
dblog [ <options> ] <database-file>
```

## Parameters

### options

#### **-ek key**

Specify the encryption key for strongly encrypted databases directly in the command. If you have a strongly encrypted database, you must provide the encryption key to use the database or transaction log in any way. For strongly encrypted databases, you must specify either `-ek` or `-ep`, but not both. The command will fail if you do not specify the correct key for a strongly encrypted database.

#### **-ep**

Specify that you want to be prompted for the encryption key. This option causes a window to appear, in which you enter the encryption key. It provides an extra measure of security by never allowing the encryption key to be seen in clear text. For strongly encrypted databases, you must specify either `-ek` or `-ep`, but not both. The command will fail if you do not specify the correct key for a strongly encrypted database.

#### **-g n**

Use this option if you are using the Log Transfer Manager to participate in a Replication Server installation. It can be used after a backup is restored, to set the generation number.

#### **-ft timeline**

Resets the timeline to `<timeline>`, so that the database can take part in replication. Use this option to reload SQL Remote consolidated databases. Specify the timeline in the format: "**<GUID>**"

`<timestamp>`", where `<timestamp>` is a UTC representation of time in the format "YYYY-MM-DD HH:MM:SS.MMMM", which is the same format used in dblog output.

**-il**

Use this option if you have stopped using the Log Transfer Manager to participate in a Replication Server installation on this database, but continue to use SQL Remote or MobiLink synchronization. It resets the Log Transfer Manager log offset that is kept for the `delete_old_logs` option, allowing transaction logs to be deleted when they are no longer needed.

**-is**

Use this option if you have stopped using MobiLink synchronization on this database, but continue to use the Log Transfer Manager or SQL Remote. It resets the MobiLink log offset that is kept for the `delete_old_logs` option, allowing transaction logs to be deleted when they are no longer needed.

**-m mirror-name**

Specify the file name for a new transaction log mirror. If the database is not currently using a transaction log mirror, it starts using one. If the database is already using a transaction log mirror, it changes to using the new file as its transaction log mirror.

**-n**

Stop using a transaction log, and stop using a transaction log mirror. Without a transaction log, the database can no longer participate in data replication or use the transaction log in data recovery. If a SQL Remote, Log Transfer Manager, or `dbmlsync truncation offset` exists, the transaction log cannot be removed unless the corresponding ignore option (`-il` for the Log Transfer Manager, `-ir` for SQL Remote, or `-is` for `dbmlsync`) is also specified. You cannot stop using a transaction log if the database has auditing turned on (unless you first turn auditing off).

**-o filename**

Write output messages to the named file.

**-q**

Run in quiet mode; messages are not displayed.

**-r**

Maintain a single transaction log for databases that maintain a transaction log mirror.

**-t log-name**

Specify the file name for a new transaction log. If the database is not currently using a transaction log, it starts using one. If the database is already using a transaction log, it changes to using the new file as its transaction log.

**-x n**

Reset the transaction log current relative offset to `n`, so that the database can take part in replication. This option is used for reloading SQL Remote consolidated databases.

**-z n**

Reset the transaction log starting offset to `n`, so that the database can take part in replication. This option is used for reloading SQL Remote consolidated databases.

#### **database-file**

The name of the database file.

## Privileges

None.

## Remarks

The `dblog` command line utility allows you to display or change the name of the transaction log or transaction log mirror. You can also stop or start maintaining a transaction log mirror.

The name of the transaction log is set when you create a database. The database cannot be running when you change its transaction log file name.

When you use the `RESTORE DATABASE` statement to move and/or rename a database, you can rename all of the files except the transaction log. Transactions continue to be written to the old log file name, in the location where the catalog store file (the `.db` file) is located after the database is restored.

When you rename or move all other files in the database, do the same for the log file. Use `dblog` to move or rename the log file. Run this utility after using `RESTORE DATABASE` with the following:

- A new database name
- The `RENAME` option

You can use `dblog` to rename the transaction log even if you have not restored the database, given these restrictions:

- The SAP IQ server must be stopped.
- After the log is renamed, retain the old log until the next database backup, in case the old log is needed for recovery from a media failure.

`dblog` displays additional information about the transaction log:

- Version number
- Starting offset, for use in replication
- Ending offset, for use in replication
- Page size
- Total number of pages
- Number of empty pages

# 12 dbping Database Administration Utility

The `dbping` utility is a ping utility that assists in diagnosing connection problems.

## ↳ Syntax

```
dbping [ <options> ]
```

## Parameters

### options

#### **-c connection-string**

Supply database connection parameters, in the form of "`<keyword>=<value>; ...`".

See *Connection and Communication Parameters Reference* in *SAP IQ Administration: Database*.

If no connection parameters are specified, connection parameters from the `SQLCONNECT` environment variable are used, if set.

#### **-d**

Make a database connection if the server is found.

If you do not supply the `-d` option, `dbping` reports success if it finds the server specified by the `-c` option. If you do supply the `-d` option, `dbping` reports success only if it connects to both server and database.

For example, if you have a server named `blair` running the `iqdemo` database, this command succeeds:

```
dbping -c "eng=blair;dbn=iqdemo"
```

The following command fails, with the message `Ping database failed -- specified database not found`:

```
dbping -d -c "eng=blair;dbn=iqdemo"
```

#### **-en**

Specifies that you want `dbping` to exit with a failed return code when `NULL` is returned for any of the properties specified. By default, `dbping` prints `NULL` when the value for a property specified by `-pc`, `-pd`, or `-ps` is unknown, and exits with a success return code. This option can only be used with `-pc`, `-pd`, and `-ps`.

#### **-l library**

Use the specified ODBC driver or driver manager library. Specify the library to use (without its file extension). This option does not use the ODBC Driver Manager, and so is particularly useful on UNIX operating systems.

For example, this command directly loads the ODBC driver:

```
dbping -m -c "dsn=IQ16iqdemo" -l dbodbc11
```

Use `dbping` to verify connectivity with the ODBC Driver Manager on UNIX systems. Unlike Interactive SQL or other tools, `dbping` allows you to explicitly test the components as you would with a third-party tool. In the following example, `libodbc.so` is the third-party ODBC driver:

```
dbping -m -c "dsn=dsnname" -l /<full path>/libodbc.so
```

#### **-m**

Use the ODBC Driver Manager. Otherwise, connect using Embedded SQL. Establish a connection using ODBC. By default, `dbping` attempts a connection using the embedded SQL interface.

#### **-o filename**

Log output messages to a file.

#### **-pc property,...**

Upon connection, display the specified connection properties. Supply the properties in a comma-separated list. You must specify enough connection information to establish a database connection.

#### **-pd property,...**

Upon connection, display the specified database properties. Supply the properties in a comma-separated list. You must specify enough connection information to establish a database connection.

#### **-ps property,...**

Upon connection, display the specified database server properties. Supply the properties in a comma-separated list.

#### **-q**

Operate quietly; do not print messages. If `dbping` fails, a message always appears.

#### **-s**

Returns information about the performance of the network between the computer running `dbping` and the computer running the database server. Approximate connection speed, latency, and throughput are displayed. The `-c` option is usually required to specify the connection parameters to connect to a database on the server. You can only use `dbping -s` for embedded SQL connections. This option is ignored if `-m` or `-l` is also specified. By default, `dbping -s` loops through the requests for at least one second for each statistic it measures. A maximum of 200 connect and disconnect iterations are performed, regardless of the time they take, to avoid consuming too many resources. On slower networks, it can take several seconds to perform the minimum number of iterations for each statistic. The performance statistics are approximate, and are more accurate when both the client and server computers are fairly idle.

#### **-st time**

This option is the same as `-s`, except that it specifies the length of time, in seconds, that `dbping` loops through the requests for each statistic it measures. This option allows more accurate timing information to be obtained than `-s`.

#### **-z**

Display debugging information. This option is available only when an embedded SQL connection is being attempted. That is, it cannot be combined with `-m` or `-l`. It displays the network communication protocols used to attempt connection, and other diagnostic messages.

## Privileges

None.

## Remarks

Use `dbping` to help debug connection problems. You can enter a full or partial connection string; the utility returns a message indicating whether the attempt to locate a server or database, or to connect, was successful.

Use `dbping` for Embedded SQL or ODBC connections. You cannot use `dbping` for jConnect (TDS) connections.

Exit codes are 0 (success) or nonzero (failure).

This utility accepts `@filename` parameters.

See *Connection and Communication Parameters Reference* in *SAP IQ Administration: Database*.

# 13 dbstop Database Shutdown Utility

`dbstop` stops a database server. The `dbstop` utility is a command line utility only, available on both UNIX and Windows platforms.

In UNIX, `dbstop` can shut down a server on any node on the network. You must specify a `<server-name>`, as well as any connection parameters you specified when you started the server. Without the proper connection parameters, `dbstop` does not know how to connect to the server to shut it down.

The `dbstop` command-line options control whether a server is stopped, even if there are active connections.

`SERVER` is the default logical server for `dbstop`.

## ⌘ Syntax

```
dbstop [ <options> ] <server-name>
```

## Parameters

### options

#### @filename

Read in options from the specified environment variable or configuration file.

#### -c "keyword=value; ..."

To stop a network server, supply a connection string with a user ID that has permissions to stop the server. By default, the `SERVER OPERATOR` system privilege is required on the network server, and all users can shut down a personal server; you can use the `-gk` server command-line option to change the default behavior.

See *Connection and Communication Parameters Reference* in *Administration: Database*.

To stop a multiplex server, you need permissions to stop the server in the logical server context in which you connect to it. By default, `dbstop` logs in with `SERVER` context, so you would need permission to log into the `SERVER` logical server.

If there are active connections, `dbstop` prompts whether you want to shut down the server. If you specify `unconditional=true` on the command line, the server shuts down without prompting, even if there are active connections.

Avoid overriding the logical server in the connection string; if login redirection is enabled in the overriding user's login policy, the server may shut down as a connection is redirected.

#### -d

Does not stop the database server. Instead, only stops the database specified in the connection string.

#### -o filename

Logs output messages to the named file.

**-q**

Quiet mode; does not print messages.

**-x**

Does not stop if there are active connections. Including this option prevents `dbstop` from prompting for confirmation if there are active connections.

**-y**

Stops without prompting even if there are active connections.

**server-name**

Server name of a running server to stop. If you supply a server name, do not supply connection parameters as well.

## Examples

- This example stops a database named `iqdemo` on the server `myserver`, enter:

```
dbstop -c "uid=DBA;pwd=<password>;eng=myserver;dbn=iqdemo"
```

- This example stops a server, `myserver`, regardless of the database running:

```
dbstop -c "uid=DBA;pwd=<password>;eng=myserver;dbn=utility_db"
```

# 14 dbtran Database Administration Utility

Use the `dbtran` log translation utility, at the command prompt, to translate a transaction log into a `.sql` command file.

## ↳ Syntax

### Syntax 1 – Running Against a Database Server

```
dbtran [ <options> ] -c { <connection-string> } -n <filename>
```

### Syntax 2 – Running Against a Transaction Log

```
dbtran [ <options> ] [ <transaction-log> ] [ <SQL-file> ]
```

## Parameters

### options

#### @data

Reads in options from the specified environment variable or configuration file.

#### -a

Controls whether uncommitted transactions appear in the transaction log. The transaction log contains changes made only before the most recent `COMMIT` by any transaction. If you do not specify `-a`, only committed transactions appear in the output file. If you specify `-a`, any uncommitted transactions found in the transaction log appear.

#### -d

Specifies that transactions are written in order from earliest to latest. This feature is intended for auditing database activity: do not apply `dbtran` output against a database.

#### -ek key

Specifies the encryption key for strongly encrypted databases. If you have a strongly encrypted database, you must provide the encryption key to use the database or transaction log. Specify either `-ek` or `-ep`, but not both. The command fails if you do not specify the correct encryption key. If you are running `dbtran` against a database server using the `-c` option, specify the key using a connection parameter instead of using the `-ek` option. For example, the following command gets the transaction log information about database `enc.db` from the database server `sample`, and saves its output in `log.sql`:

```
dbtran -c "ENG=sample;DBF=enc.db;UID=DBA;PWD=<password>;DBKEY=mykey" -n log.sql
```

#### -ep

Prompts for the encryption key. This option causes a window to appear, in which you enter the encryption key. It provides an extra measure of security by never allowing the encryption key to be

seen in clear text. Specify either `-ek` or `-ep`, but not both. The command fails if you do not specify the correct encryption key. If you are running `dbtran` against a database server using the `-c` option, specify the key using a connection parameter, instead of using the `-ep` option. For example, the following command gets the transaction log information about database `enc.db` from the database server `sample`, and saves its output in `log.sql`.

```
dbtran -c "ENG=sample;DBF=enc.db;UID=DBA;PWD=<password>;DBKEY=mykey" -n
log.sql
```

#### **-f**

Outputs only transactions completed since the last checkpoint.

#### **-g**

Adds auditing information to the transaction log if the auditing database option is turned on.

#### **-ir offset1,offset2**

Outputs a portion of the transaction log between two specified offsets.

#### **-is source, ...**

Outputs operations on rows that have been modified by operations from one or more of the following sources, specified as a comma-separated list:

- `All` – all rows. This is the default setting.
- `SQLRemote` – include only rows that were modified using SQL Remote. You can also use the short form "SR".
- `RepServer` – include only rows that were modified using the Replication Agent (LTM) and Replication Server. You can also use the short form "RS".
- `Local` – include only rows that are not replicated.

#### **-it owner.table, ...**

Outputs operations on the specified, comma-separated list of tables. Specify each table as `owner.table`.

#### **-j date/time**

Translates only transactions from the most recent checkpoint prior to the given date or time. The user-provided argument can be a date, time, or date and time, enclosed in quotes. If you omit a time, the default is 00:00. If you omit a date, the current day is the default. The acceptable format for the date and time is: "YYYY/MMM/DD HH:NN".

#### **-k**

Prevents partial `.sql` files from being erased if an error is detected. If an error is detected while `dbtran` is running, the `.sql` file generated until that point is normally erased to ensure that a partial file is not used. Specifying this option may be useful if you are attempting to salvage transactions from a damaged transaction log.

#### **-m**

Specifies a directory that contains transaction logs. Use this option with the `-n` option.

#### **-o filename**

Writes output messages to the named file.

#### **-r**

Removes any uncommitted transactions. This is the default behavior.

**-rsu username, ...**

Specifies a comma-separated list of user names to override the default Replication Server user names. By default, the `-is` option assumes the default Replication Server user names of `dbmaint` and `sa`.

**-s**

Controls how `UPDATE` statements are generated. If you do not use this option, and there is no primary key or unique index on a table, `dbtran` generates `UPDATE` statements with a nonstandard `FIRST` keyword in case of duplicate rows. If you do use this option, the `FIRST` keyword is omitted for compatibility with the SQL standard.

**-sr**

Includes in the output file generated comments describing how SQL Remote distributes operations to remote sites.

**-t**

Controls whether triggers are included in the command file. By default, actions performed by triggers are not included in the command file. If the matching trigger is in the database, when the command file is run against the database, the trigger performs the actions automatically. Trigger actions should be included if the matching trigger does not exist in the database against which the command file is to run.

**-u userid, ...**

Limits the output from the transaction log to include only specified users.

**-x userid, ...**

Limits the output from the transaction log to exclude specified users.

**-y**

Replaces existing command files without prompting for confirmation. If you specify `-q`, you must also specify `-y` or the operation fails.

**-c connection-string**

Specifies the connection string when running the utility against a database server, and takes the form of "`<keyword>=<value>; ...`".

**-n filename**

Specifies the output file that holds the SQL statements when you run `dbtran` against a database server.

**transaction-log**

Specifies the log file to be translated. Cannot be used with `-c` or `-m` options.

**SQL-file**

Names the output file containing the translated information. For use with `<transaction-log>` only.

## Remarks

You can run `dbtran` against the following:

- Against a database server – connects to the database server using the connection string specified following the `-c` option, and places output in a file specified with the `-n` option. The `BACKUP DATABASE`

system privilege is required to run in this way. For example, this command translates log information from the `iqdemo` server and places the output in a file named `iqdemo.sql`:

```
dbtran -c "eng=iqdemo;dbn=iqdemo;  
dbf=iqdemo.db;uid=DBA;pwd=MyIQD@tabase" -n iqdemo.sql
```

- Against a transaction log file – acts directly against a transaction log file. Protect your transaction log file from general access to prevent users from running this statement.

```
dbtran iqdemo.log iqdemo.sql
```

`dbtran` shows the earliest log offset in the transaction log, which you can use to determine the order in which multiple log files were generated.

`dbtran-c` attempts to translate the online transaction log file, and all the offline transaction log files in the same directory as the online transaction log file. If the directory contains transaction log files for more than one database, you may see an error. To avoid this, ensure that each directory contains transaction log files for only one database.

A transaction can span multiple transaction logs. If transaction log files contain transactions that span logs, translating a single transaction log file (for example, `dbtran <demo.log>`) might lose the spanning transactions. For `dbtran` to generate complete transactions, use the `-c` or `-m` options with the transaction log files in the directory.

Exit codes are 0 (success) or nonzero (failure).

This utility accepts `@filename` parameters.

# 15 dbvalid Database Administration Utility

The `dbvalid` utility is a validation utility that validates the indexes and keys on some or all of the SAP SQL Anywhere tables in the catalog store.

The utility scans the entire table and looks up each record in every index and key defined on the table. By default, the utility uses the express check option.

## Note

The `dbvalid` utility lets you easily validate catalog store tables, but does not validate IQ main store tables. Use the `sp_iqcheckdb` stored procedure to validate IQ main store tables.

You can access the `dbvalid` utility at the system command-line level, which is useful for incorporating `dbvalid` into batch or command files.

## Syntax

```
dbvalid [ <options> ] [ <object-name>,... ]
```

## Parameters

### options

#### **-c "keyword=value; ..."**

Supply database connection parameters. The user ID must have the `VALIDATE ANY OBJECT` system privilege.

For example, the following validates the `iqdemo` database, connecting as user `DBA` with password `sql`:

```
dbvalid -c "uid=DBA;  
pwd=<password>;  
dbf-/IQ/IQ-16_1/demo/iqdemo.db"
```

#### **-o filename**

Log output messages to a file

#### **-f**

Validate tables with full check. In addition to the default validation checks, carry out both data checks (`-fd`) and index checks (`-fi`). This option corresponds to the `WITH FULL CHECK` option of the SQL Anywhere `VALIDATE TABLE` statement. Depending on the contents of your catalog store, this option may significantly extend the time required to validate.

#### **-fd**

Validate tables with data check. In addition to the default validation checks, check that all of each `LONG BINARY`, `LONG VARCHAR`, `TEXT` or `IMAGE` data type can be read. Entries with these data types may span more than one page. In the catalog store:

- Domain – user-defined data type.
- IMAGE – a domain to LONG BINARY
- TEXT – a domain to LONG VARCHAR

This option instructs the database server to check all pages used by each entry. This corresponds to the WITH DATA CHECK option on the SQL Anywhere VALIDATE TABLE statement. Depending on the contents of your catalog store, this option may significantly extend the time required to validate.

**-fi**

Validate tables with index check. In addition to the default validation checks, validate each index on the table. This corresponds to the WITH INDEX CHECK option of the SQL Anywhere VALIDATE TABLE statement. Depending on the contents of your catalog store, this option may significantly extend the time required to validate.

**-fx**

Validate tables with express check. In addition to the default and data checks, check that the number of rows in the table matches the number of entries in the index. This corresponds to the WITH EXPRESS CHECK of the SQL Anywhere VALIDATE TABLE statement. This option does not perform individual index lookups for each row.

**-i**

Each <object-name> is an index. Instead of validating tables, validate indexes. Ensure that every row referenced in the index actually exists in the table. For foreign-key indexes, -i also ensures that the corresponding row exists in the primary table. If you supply a <table-name> instead of an <index-name,> validates the primary key index. In this case, for dbvalid, each of the <object-name >values supplied represents an index instead of a table and has a name in the following format:

```
[ [ owner . ] table-name . ] index-name
```

Must be the owner of the table on which the index is created or have the VALIDATE ANY OBJECT system privilege.

**-q**

Operate quietly — do not print output messages.

**-s**

Validate database pages using checksums. Checksums are used to determine whether a database page has been modified on disk. If you created a database with checksums enabled, you can validate the catalog store using checksums. Checksum validation reads each page of the catalog store from disk and calculates its checksum. If the calculated checksum is different from the checksum stored on the page, the page has been modified on disk and an error is returned. The page numbers of any invalid catalog store pages appear in the server messages window. You cannot use -s with -i, -t, or any of the -f options.

**-t**

The list of <object-name> values is a list of tables. This is also the default behavior.

**object-name**

The name of a table or (if -i is used) an index to validate

## Remarks

With the `dbvalid` command line utility, you can validate the indexes and keys on some or all of the tables in the catalog store. `dbvalid` scans the entire table and confirms that each row exists in the appropriate indexes. It is the same as running the SQL Anywhere `VALIDATE TABLE` statement on each catalog store table.

### Note

`VALIDATE TABLE` is not supported in SAP IQ. `sp_iqcheckdb` provides a similar function for IQ main store tables.

By default, the Validation utility uses the express check option. However, the express check option is not used if you specify `-f`, `-fd`, `-fi`, `-fn`, or `-i`.

If the catalog store table is inconsistent, `dbvalid` reports an error. If errors are reported, you can drop all of the indexes and keys on a table and re-create them. You must also re-create any foreign keys to the table.

### Caution

Validate a table or entire catalog store only when no connections are making changes to the database; otherwise, spurious errors may be reported, indicating some form of database inconsistency even though no inconsistency actually exists.

The program exit codes and their descriptions are as follows:

- 0 – database validated successfully
- 1 – general failure in utility
- 2 – error validating database
- 7 – cannot find database to connect to (database name is wrong)
- 8 – cannot connect to database (user ID/password is wrong)
- 11 – cannot find server to connect to (server name is wrong)
- 12 – incorrect encryption key for starting database

## Example

The following command validates the catalog store of the `iqdemo` database, connecting as user `DBA` with password `sql`:

```
dbvalid -c "uid=DBA;pwd=<password>;dbf-/IQ/IQ-16_1/demo/iqdemo.db"
```

# 16 dsedit Server Administration Utility

The `dsedit` utility allows you to view and edit server entries in the interfaces file using a GUI. In Windows, `dsedit` creates and modifies network connection information in the interfaces file.

The utility is located in:

- (UNIX) `$SYBASE/$SYBASE_OCS/bin`.
- (Windows) `%SYBASE%\%SYBASE_OCS%\bin`, as `dsedit.exe`.

## Syntax

### Syntax 1

```
dsedit
```

### Syntax 2

```
dsedit -v
```

## Parameters

`-v`

Displays the version and copyright message of `dsedit`.

## Remarks

Set the SYBASE environment variable:

- To the location of the current version of SAP ASE before using `dsedit`.
- Before invoking `dsedit`, unless you are only using the `-v` parameter to display the version number.

Once you are in an open session, you can add, modify, rename, and delete server entries associated with that session, as well as copy server entries within a session and between sessions. Each server entry consists of a set of attributes:

Attribute name	Type of value	Description	Default value
Server Version	Integer	Version level of the server object definition. SAP provides this attribute to identify future changes to the object definition.	150
Server Name	Character string	Server name.	N/A

Attribute name	Type of value	Description	Default value
Server Service	Character string	A description of the service provided by the server. This value can be any meaningful description.	The SAP IQ server
Server Status	Integer	The operating status of the server. Values are: <ul style="list-style-type: none"> <li>Active</li> <li>Stopped</li> <li>Failed</li> <li>Unknown</li> </ul>	4
Security Mechanism	Character string	(Optional) Object identifier strings (OID) that specify the security mechanisms supported by the server. If it is not specified, Open Server allows clients to connect with any security mechanism for which Open Server has a corresponding security driver.	N/A
Server Address	Character string	One or more addresses for the server.  The format of the address varies by protocol, and some protocols allow more than one format. The options are: <ul style="list-style-type: none"> <li>TCP/IP – two formats: <ul style="list-style-type: none"> <li><code>&lt;computer name&gt;,&lt;port number&gt;</code></li> <li><code>&lt;ip-address&gt;,&lt;portnumber&gt;</code></li> </ul> </li> <li>Named Pipe – <code>&lt;pipe name&gt;:"\pipe"</code> is a required prefix to all pipe names. Server pipes can be only local. <ul style="list-style-type: none"> <li>Local – <code>\pipe\sql\query</code></li> <li>Remote – <code>\\computer_name\pipe\sql\query</code></li> </ul> </li> <li>IPX/SPX – three formats: <ul style="list-style-type: none"> <li><code>&lt;server name&gt;</code></li> <li><code>&lt;net number&gt;,&lt;node number&gt;,&lt;socket number&gt;</code></li> <li><code>&lt;server name&gt;,&lt;socket number&gt;</code></li> </ul> </li> <li>DECnet – four formats: <ul style="list-style-type: none"> <li><code>&lt;area number&gt;.&lt;node number&gt;,&lt;object name&gt;</code></li> <li><code>&lt;area number&gt;.&lt;node number&gt;,&lt;object number&gt;</code></li> <li><code>&lt;node name&gt;,&lt;object name&gt;</code></li> <li><code>&lt;node name&gt;,&lt;object number&gt;</code></li> </ul> </li> </ul>	N/A

### In this section:

#### [Using dsedit in UNIX \[page 52\]](#)

The `dsedit` utility allows you to view and edit server entries in the interfaces file using a GUI.

#### [Using dsedit in Windows \[page 53\]](#)

The `dsedit` utility allows you to view and edit server entries in the interfaces file using a GUI. In Windows, `dsedit` creates and modifies network connection information in the interfaces file.

#### [Troubleshoot dsedit \[page 55\]](#)

Identify and correct common `dsedit` problems.

## 16.1 Using dsedit in UNIX

The `dsedit` utility allows you to view and edit server entries in the interfaces file using a GUI.

### Prerequisites

If you are running `dsedit` from a remote machine, verify that the `DISPLAY` environment variable is set so the `dsedit` screens display on your machine instead of on the remote machine by logging in to the remote machine and entering:

```
setenv DISPLAY <your_machine_name>:0.0
```

The server entries associated with the session appear in the Server box. Click a server entry to select it.

### Procedure

1. Start `dsedit` by entering:

```
$SYBASE/bin/dsedit
```

The Select a Directory Service window appears, letting you open editing sessions for the interfaces file. The full path name of the default interfaces file is shown in the Interfaces File to Edit box, and the full path name of the configuration file is shown below it.

2. Open a Directory Service Session window by selecting **Sybase Interfaces File**:
  - To open the default interfaces file for editing, click **OK**.
  - To open a file other than the default interfaces file, edit the displayed file name, then click **OK**.
3. The server entries associated with the session appear in the Server box. Click a server entry to select it, and choose:

Option	Description
To add a new server entry	Select <b>Add new server entry</b> to specify the name and network addresses for a new server entry.
To modify a server entry	Select <b>Modify server entry</b> to view the server entry, or change its attributes.
To copy a server entry within the current session	<ol style="list-style-type: none"><li>1. Select one or more servers to copy. To select multiple entries, use the <code>Shift</code> key.</li><li>2. Click <b>Copy</b> from below the menu bar, or choose <b>Edit &gt; Copy</b>.</li><li>3. Click <b>Paste</b> from below the menu bar, or choose <b>Edit &gt; Paste</b>. <code>dsedit</code> appends the copied server entries with a version number of <code>&lt;_n&gt;</code>. To rename copied server entries, select <b>Server Object &gt; Rename</b>.</li></ol>

Option	Description
<b>To copy a server entry to another interfaces file</b>	<ol style="list-style-type: none"> <li>1. Select the entry to copy, and click once. To copy a range of consecutive entries, select the first entry in the range, then press and hold the <code>Shift</code> key, and select the last entry in the range. To select multiple, nonconsecutive entries, press and hold down the <code>Ctrl</code> key while you click each entry.</li> <li>2. Click <b>Copy server entry</b>.</li> <li>3. Select the interfaces file from the list.</li> <li>4. Edit the file name, then click <b>OK</b>.</li> </ol>
<b>To copy server entries between sessions</b>	<ol style="list-style-type: none"> <li>1. Select one or more servers to copy. To select multiple entries, use the <code>Shift</code> key.</li> <li>2. Click <b>Copy</b> from below the menu bar, or choose <b>Edit &gt; Copy</b>. To cut the server entries, click <b>Cut</b> below the menu bar, or choose <b>Edit &gt; Cut</b>.</li> <li>3. Activate the session where you want to paste the server entries.</li> <li>4. Click <b>Paste</b> below the menu bar, or choose <b>Edit &gt; Paste</b>.</li> </ol> <p>You can rename the copied server entries using <b>Server Object &gt; Rename</b>.</p>

4. View, edit, or create the transport addresses at which a server accepts client connections in the Network Transport Editor window. The window lets you configure:
  - Transport type – specifies the protocol and interface for the address.
  - Address information – different address components are required depending on the transport type.

## 16.2 Using dsedit in Windows

The `dsedit` utility allows you to view and edit server entries in the interfaces file using a GUI. In Windows, `dsedit` creates and modifies network connection information in the interfaces file.

### Prerequisites

The server entries associated with the session appear in the Server box. Click a server entry to select it.

### Procedure

1. Start `dsedit` using one of these methods:

Option	Description
Command prompt	<p>Enter:</p> <pre>dsedit</pre> <p>Specify these command line arguments:</p> <ul style="list-style-type: none"> <li>• <code>-d&lt;dsname&gt;</code> – specifies which directory service to connect to. <code>&lt;dsname&gt;</code> is the local name of the directory service, as listed in the <code>libtcl.cfg</code> file. If you do not specify <code>-d&lt;dsname&gt;</code>, <code>dsedit</code> presents a list of directory service options in the first dialog box.</li> <li>• <code>-l&lt;path&gt;</code> – specifies the path to the <code>libtcl.cfg</code> file, if other than <code>SYBASE_home\INI</code>. Use this only if you want to use a <code>libtcl.cfg</code> file other than the one located in <code>SYBASE_home\INI</code>.</li> </ul>
Windows Explorer	Double-click <code>DSEEDIT.exe</code> in the <code>%SYBASE%\bin\</code> directory.
Windows Start menu	Choose <b>Start &gt; Sybase for Windows &gt; dsedit</b> .

The Select Directory Service dialog appears, letting you open editing sessions for the interfaces file.

- Open a session with a directory service through the Select Directory Service dialog, using one of these files:
  - Any directory service that has a driver listed in the `libtcl.cfg` file. `dsedit` uses the SYBASE environment variable to locate the `libtcl.cfg` file. If the SYBASE environment variable is not set correctly, `dsedit` cannot locate the `libtcl.cfg` file.
  - The `sql.ini` file.

Click the local name of the directory service you want to connect to, as listed in the DS Name box, and click **OK**.

The session number and local name of the directory service appear in the header bar.

- The server entries associated with the session appear in the Server box. Click a server entry to select it, and choose:

Option	Description
To add a new server entry	<ol style="list-style-type: none"> <li>Select a server and choose <b>Server Object &gt; Add</b>.</li> <li>Enter a new server name, then click <b>OK</b>.</li> </ol>
To modify a server entry	<ol style="list-style-type: none"> <li>Select a server and choose <b>Server Object &gt; Modify Attribute</b>.</li> <li>Select the attribute to modify. A dialog box appears that shows the current value of the attribute.</li> <li>Enter or select the new attribute value, then click <b>OK</b>.</li> </ol>
To rename a server entry	<ol style="list-style-type: none"> <li>Select a server and choose <b>Server Object &gt; Rename</b>.</li> <li>Enter a new server name, then click <b>OK</b>.</li> </ol>

Option	Description
<b>To copy a server entry within the current session</b>	<ol style="list-style-type: none"> <li>1. Select one or more servers to copy. To select multiple entries, use the <code>Shift</code> key.</li> <li>2. Click <b>Copy</b> from below the menu bar, or choose <b>Edit &gt; Copy</b>.</li> <li>3. Click <b>Paste</b> from below the menu bar, or choose <b>Edit &gt; Paste</b>. <code>dsedit</code> appends the copied server entries with a version number of <code>&lt;_n&gt;</code>. To rename copied server entries, select <b>Server Object &gt; Rename</b>.</li> </ol>
<b>To copy a server entry to another interfaces file</b>	<ol style="list-style-type: none"> <li>1. Select the entry to copy, and click once. To copy a range of consecutive entries, select the first entry in the range, then press and hold the <code>Shift</code> key, and select the last entry in the range. To select multiple, nonconsecutive entries, press and hold down the <code>Ctrl</code> key while you click each entry.</li> <li>2. Click <b>Copy server entry</b>.</li> <li>3. Select the interfaces file from the list.</li> <li>4. Edit the file name, then click <b>OK</b>.</li> </ol>
<b>To copy server entries between sessions</b>	<ol style="list-style-type: none"> <li>1. Select one or more servers to copy. To select multiple entries, use the <code>Shift</code> key.</li> <li>2. Click <b>Copy</b> from below the menu bar, or choose <b>Edit &gt; Copy</b>. To cut the server entries, click <b>Cut</b> below the menu bar, or choose <b>Edit &gt; Cut</b>.</li> <li>3. Activate the session where you want to paste the server entries.</li> <li>4. Click <b>Paste</b> below the menu bar, or choose <b>Edit &gt; Paste</b>.</li> </ol> <p>You can rename the copied server entries using <b>Server Object &gt; Rename</b>.</p>
<b>To delete a server entry</b>	Select a server and choose <b>Server Object &gt; Delete</b> .

## 16.3 Troubleshoot dsedit

Identify and correct common `dsedit` problems.

### The dsedit Utility Does Not Start

Check for the following:

- The SYBASE environment variable is not set, or points to the wrong directory.
- (UNIX) X-Windows is not configured correctly. If you are running `dsedit` on a remote host, make sure that X-Windows clients on the remote host can connect to the X-Windows server on your own machine. See your X-Windows documentation for more troubleshooting information. If X-Windows is not available, use `dscp` instead of `dsedit`.

## Error Message: "Unable to Open X Display"

(UNIX) `dsedit` might not work if the display machine is set up to reject X-Windows connections from remote hosts. If this is the problem, you see a message similar to the following:

```
Unable to open X display. Check the value of your
$DISPLAY variable. If it is set correctly, use the
'xhost +' command on the display machine to authorize
use of the X display. If no X display is available, run
dscp instead of dsedit.
```

This error may be caused by either of the following situations:

- The value for the DISPLAY environment variable is not entered correctly or is not set.  
**Solution:** Enter the DISPLAY environment variable correctly.
- You are not authorized to open windows on the machine to which DISPLAY refers.  
**Solution:** Run the command `xhost +` on the display machine.

## Cannot Add, Modify, or Delete Server Entries

Check for permissions problems with the interfaces file. To edit interfaces entries, you must have write permission on both the interfaces file and the Sybase installation directory.

# 17 iqbackupcheck Backup Verification Utility

The `iqbackupcheck` utility internally runs the `restore` statement with the `verify` clause to check whether or not a backup is good to be restored.

## ↔ Syntax

```
iqbackupcheck -a <backup archive path and name> [<options>]
```

## Parameters

- a (Mandatory) An indicator that precedes the path/name of the backup archive file.
- b An indicator that precedes the name of the BACKINT parameter file – this is required when performing BACKINT-based backups.
- e An indicator that precedes the external backup ID (EBID).
- i An indicator that precedes the backup ID.
- p An indicator that precedes the path/name of the log file.

## Privileges

None.

## Remarks

- The verification process reports errors similar to the way `restore database` does – the only difference being no writes to the database with `iqbackupcheck`.
- A checksum is run against the backup archive to ascertain the validity and consistency of the backup.
- Verification progress reporting**  
The `RESTORE VERIFY` process verifies every stripe specified in the command. As the verification process checks stripes and their corresponding files, it reports progress in terms of the number of IQ blocks verified. After every 5000 IQ blocks verified, a message is written to the server log file. Example:  
5000/100000 (5%) Blocks verified.  
A final message is written to the server log file when 100% of the IQ blocks are verified. Messages indicating the start of the `RESTORE VERIFY` process, the number of IQ blocks to be verified, and the completion of the `RESTORE VERIFY` are also written to the server log. Example:

```
11/17 06:45:24. VERIFY RESTORE Started  
11/17 06:45:24. Total number of IQ blocks to be verified: 764
```

```
11/17 06:45:24. Total number of IQ blocks verified: 764/764 (100%)
11/17 06:45:24. VERIFY RESTORE Successfully Completed
```

- **Verification error reporting**

If the verification process finds errors it can skip, it continues to check the archive and log the errors it detects. The errors that can be skipped and allow for the verification to continue (but not succeed) are:

- Header of the block to be restored appears to be corrupted. (SQLCODE -10120111, SQLSTATE QUA11)
- Media data appears corrupted (bad checksum). (SQLCODE -1012012, SQLSTATE QUA12)
- Media meta data appears corrupted (boundary record). (SQLCODE -1012013, SQLSTATE QUA13)
- Media meta data appears corrupted (multiple begin boundary records). (SQLCODE -1012014, SQLSTATE QUA14)

If any of the errors above are encountered but the verification process continues to the end of the archive, the verification fails and SAP IQ reports the following error:

```
The verification of the provided archive has failed. Please check the server
log for details of the errors thrown during verify.
iqbackupcheck failed!
```

If any error pertaining to RESTORE other than the errors above is encountered, that particular error is reported, and the verification process stops.

## Examples

- File-based backup:

```
iqbackupcheck -a abc_databackup_1024_1_1543764015386
```

In this example, `abc_databackup_1024_1_1543764015386` indicates the name of the backup archive.

- BACKINT-based backup:

```
iqbackupcheck -a testbkup_databackup_1024_1_1542185723522 -b BCKINTCQbxUi
```

In this example, `testbkup_databackup_1024_1_1542185723522` indicates the name of the backup archive, and `BCKINTCQbxUi` indicates the BACKINT parameter file.

# 18 iqdscp Configuration Utility

The `iqdscp` utility is a UNIX-only Open Client and Open Server configuration utility you can use to configure the interfaces file and to configure a directory service.

See [Open Server 15.5 > Open Client and Open Server Configuration Guide for UNIX > Using dscp](#) .

## 19 (Deprecated) iqsedit Utility

All support for the `iqsedit` utility has been discontinued. Use the `dsedit` utility (`dsedit.exe`) instead.

## 20 iqdsn Database Administration Utility

The `iqdsn` utility is a data source utility you use for creating, changing, deleting, describing, and listing ODBC data sources.

### Syntax

```
iqdsn [ <modifier-options> ]
{
  -l
  -d <dsn>
  -g <dsn>
  -w <dsn> [<details-options>;...]
  -cl }
```

```
iqdsn [ <modifier-options> ]
{
  -l[ s | u ]
  -d[ s | u ] <dsn>
  -g[ s | u ] <dsn>
  -w[ s | u ] <dsn> [<details-options>;...]
  -cl }
```

Major option	Description
@<data>	Reads options from the specified environment variable or configuration file.  To protect information in the configuration file, you can use the File Hiding utility (dbfhide) to encode the contents of the configuration file.
-l[ s   u ]	Lists the available ODBC data sources. You can modify the list format using the -b or -v options. On Microsoft Windows, you can modify the option using the <i>u</i> (user) or <i>s</i> (system) specifiers. The default specifier is <i>u</i> .
-d[ s   u ] <dsn>	Deletes the named data source. If you supply -y, any existing data source is deleted without confirmation. On Microsoft Windows, you can modify the option using the <i>u</i> (user) or <i>s</i> (system) specifiers. The default specifier is <i>u</i> .
-g[ s   u ] <dsn>	Lists the definition of the named data source. You can modify the format of the output using the -b or -v options. On Microsoft Windows, you can modify the option using the <i>u</i> (user) or <i>s</i> (system) specifiers. The default specifier is <i>u</i> .
-w[ s   u ] <dsn> [<details-options> ]	Creates a new data source, or overwrites one if one of the same name exists. You must specify the -c option with the -w option. If you supply -y, any existing data source is overwritten without confirmation. On Microsoft Windows, you can modify the option using the <i>u</i> (user) or <i>s</i> (system) specifiers. The default specifier is <i>u</i> .

Major option	Description
<code>-cl</code>	Lists the connection parameters supported by the <code>iqdsn</code> utility.
Modifier-options	Description
<code>-b</code>	Formats the output of the list as a single line connection string.
<code>-cm</code>	<p>Displays the command used to create the data source. This option can be used to output the creation command to a file, which can be used to add the data source to another computer or can be used to restore a data source to its original state if changes have been made to it. You must specify the <code>-g</code> option or <code>-l</code> option with <code>-cm</code> or the command fails. Specifying <code>-g</code> displays the creation command for the specified data source, while specifying <code>-l</code> displays the creation command for all data sources.</p> <p>If the specified data source does not exist, the command to delete the data source is generated. For example, if the <code>mysn</code> data source does not exist on the computer, <code>iqdsn -cm -g mysn</code> would return the following command to delete the <code>mysn</code> data source:</p> <pre>iqdsn -y -du "mysn"</pre>
<code>-dr</code>	<p>Includes the <code>Driver</code> parameter when displaying data sources. When you use the <code>-cm</code> option to recreate data sources, it allows the current version of <code>iqdsn</code> to create data sources that reference a different version of the ODBC driver.</p> <p>For example, run the following command to create a version 11 data source:</p> <pre>iqdsn -y -wu "11.0 Student Sample" -c "UID=DBA;PWD=sql;...;Driver=SQL Anywhere 11"</pre> <p>When you run <code>iqdsn -cm -l</code>, <code>iqdsn</code> lists the same command without the <code>Driver=</code> parameter, which would then recreate the ODBC data source using the SAP <b>SQL Anywhere 17</b> – Oracle ODBC driver.</p> <p>However, if you run <code>iqdsn -dr -cm -l</code>, then the <code>Driver=</code> parameter is included and the data source is recreated exactly as it was created originally: using the version 11 ODBC driver.</p>
<code>-f</code>	Displays the name of the system file that is being used. This option is only available on UNIX and Linux.

Modifier-options	Description
<i>-ns</i>	<p>Specifies that the environment variable settings are used to determine the location of the system information file (named <code>.odbc.ini</code> by default). This option is also useful for determining which file is being used by <code>iqdsn</code> when there are multiple candidates for the system information file in the environment. This option is only available on UNIX and Linux.</p> <p>If you do not specify <code>-ns</code> when creating a data source, <code>iqdsn</code> also checks for the system information file in the user's home directory and the path.</p>
<i>-o &lt;filename&gt;</i>	<p>Appends output messages to the named file.</p>
<i>-or</i>	<p>Creates a data source for the SAP <b>SQL Anywhere 17 – Oracle</b> ODBC driver driver when specified with the <code>-c</code> option. For example:</p> <pre data-bbox="826 927 1366 1010">iqdsn -w MyOracleDSN -or -c Userid=DBA;Password=passwd;ServiceName=abcd;ArraySize=500;ProcResults=y</pre> <p>You can specify the <code>-cl</code> option with the <code>-or</code> option to obtain a list of the connection parameters for the SAP <b>SQL Anywhere 17 – Oracle</b> ODBC driver.</p>
<i>-pe</i>	<p>Obfuscates the password specified in the Password (PWD) connection parameter, and stores the obfuscated password in the data source using the EncodedPassword (ENP) connection parameter. The obfuscated password can be used on any computer and provides no security. This option is not recommended and is deprecated.</p>

Modifier-options	Description
<code>-pet a   c   u</code>	<p>Encodes the password specified in the Password (PWD) connection parameter, and stores the encoded password in the data source using the ENP (EncodedPassword) connection parameter.</p> <ul style="list-style-type: none"> <li><b>-pet a</b> The password is obfuscated for use on any computer. The encoded password string and its corresponding user ID can be used by anyone on any computer to initiate a connection to this database.</li> <li><b>-pet c</b> The password is encoded on a computer such that it can only be decoded on that computer or operating system version. The encoded password string and its corresponding user ID can only be used to authenticate to a database from the computer where the encoded password was created. Anyone who can log on to the computer can use the encoded password and its corresponding user ID to authenticate to a database. It cannot be used on any other or operating system version.</li> <li><b>-pet u</b> The password is encoded on a computer by a user such that it can only be decoded on that computer or operating system version by that user only. The encoded password string and its corresponding user ID can only be used to authenticate to a database from the computer where the encoded password was created by the user who created it. It cannot be decoded by other users of that computer. It cannot be decoded on any other computer or operating system version by the same or other user.</li> </ul>
<code>-q</code>	Suppresses output to the database server messages window. If you specify <code>-q</code> when deleting or modifying a data source, you must also specify <code>-y</code> .
<code>-v</code>	Formats the output of the list over several lines, as a table.
<code>-y</code>	Deletes or overwrites each data source without prompting you for confirmation. If you specify <code>-q</code> when deleting or modifying a data source, you must also specify <code>-y</code> .

Details-options	Description
<code>-c " &lt;keyword&gt; = &lt;value&gt;;..."</code>	Specifies connection parameters as a connection string.
<code>-cw</code>	Ensures that the DBF parameter (specified using <code>-c</code> ) is an absolute file name. If the value of DBF is not an absolute file name, <code>dbdsn</code> will prepend the current working directory (CWD). This option is useful because some operating systems do not have CWD information readily available in batch files.

## Privileges

None.

## Remarks

The modifier options can occur before or after the major option specification.

Any connection parameter that is stored in a DSN will not be used if the same parameter is given directly in the connection string or in the SQLCONNECT environment variable.

The Data Source utility is a cross-platform alternative to the ODBC Data Source Administrator for creating, deleting, describing, and listing ODBC data sources. The utility is useful for batch operations.

### ⚠ Caution

You should not store a password in a data source. Storing the password in plain text in an ODBC data source provides no security beyond the steps you take to prevent unauthorized access to the location on your computer where the ODBC data source is stored. It is best to prompt for a password and/or database key.

The `-pet` option stores an encoded password in a data source so that it is not in plain text. The ENP (EncodedPassword) field of the data source contains the encoded password as a hexadecimal string. The encoded string will be decoded back to plain text by the client software during a connect request. Note that this happens on the client computer before any connection attempt is made.

The `-pet c` and `-pet u` options can only be used on the computer or computer/user for which the encoded password is intended to be used.

Although `-pet u` can be used for a Microsoft Windows System data source, it may be more appropriate to use `-pet c` since this form of encoded password can be used by any user of the computer. If the system data source is created and used by only one Microsoft Windows user, then you can use `-pet u`.

### 📌 Note

On Microsoft Windows operating systems, data sources are stored in the registry or on disk. The Data Source utility does not support the use of file-based data sources using FILEDSN.

You cannot create a data source name that is longer than 32 characters.

### 📌 Note

On UNIX and Linux operating systems, data sources are held in the system information file (named `.odbc.ini` by default). When you use the Data Source utility to create or delete ODBC data sources on UNIX or Linux, the utility automatically updates the [ODBC Data Sources] section of the system information file. If you do not specify the Driver connection parameter using the `-c` option on UNIX or Linux, the Data Source utility automatically adds a Driver entry with the full path of the **SAP SQL Anywhere 17 – Oracle** ODBC driver based on the setting of the SQLANY17 environment variable.

Do not encode the system information file (`.odbc.ini`) with the File Hiding utility (`dbfhide`) on UNIX or Linux unless you will only be using SAP IQ data sources. If you plan to use other data sources (for example,

for MobiLink synchronization), then encoding the system information file may prevent other drivers from functioning properly.

Exit codes are 0 (success) or non-zero (failure).

## Examples

- This example writes a definition of the data source `newdsn` and does not prompt for confirmation if the data source already exists:

```
iqdsn -y tcpip -w newdsn -c "uid=DBA;pwd=<password>" -v
```

- Lists all known user data sources, one data source name per line:

```
iqdsn -l
```

- Lists all data sources along with their associated connection string:

```
iqdsn -l -b
```

- Reports the connection string for user data source `MyDSN`:

```
iqdsn -g MyDSN
```

- Deletes the data source `BadDSN`, but first list the connection parameters for `BadDSN` and prompt for confirmation:

```
iqdsn -d BadDSN -v
```

- Deletes the data source `BadDSN` without prompting for confirmation:

```
iqdsn -d BadDSN -y
```

- Creates a data source named `NewDSN` for the database server `MyServer`:

```
iqdsn -w NewDSN -c "uid=DBA;pwd=<password>;eng=bar "
```

If a `NewDSN` already exists, the utility asks you if you want to overwrite it.

- The following example connects to the `sample` database server. The server name `sample` overrides the previous specified value of `MyServer`:

```
iqdsn -w NewDSN -c "uid=DBA;pwd=<password>;eng=MyServer" sample
```

- Lists all connection parameter names and their aliases:

```
iqdsn -cl
```

### In this section:

#### [ODBC Connection Parameters \[page 67\]](#)

ODBC connection parameters can be used to control the behavior of SAP IQ ODBC, JDBC, and OLE DB drivers.

## 20.1 ODBC Connection Parameters

ODBC connection parameters can be used to control the behavior of SAP IQ ODBC, JDBC, and OLE DB drivers.

ODBC connection parameters are specified along with other connection parameters in **connection strings**. In a connection string, a semicolon separates each connection parameter:

```
<parameter1>=<value1>;<parameter2>=<value2>;...
```

For example:

```
UID=DBA;PWD=passwd;DescribeCursor=always;IsolationLevel=2
```

Boolean (true or false) arguments are either YES or 1 if true, or NO or 0 if false.

These connection parameters are not supported by jConnect.

These connection parameters can be set in ODBC data sources using the iqdsn Database Administration Utility or the Microsoft ODBC Data Source Administrator.

Name	Description
AutoPreCommit	When the AutoPreCommit option is set to YES (AutoPreCommit=yes), a commit is issued before each statement is executed, so that the application can always see the latest version of all database objects. The default value is NO.
ClientAutocommit	By default, automatic commits are handled by the database server for the ODBC, JDBC, and OLE DB application programming interfaces. This is called server-side autocommit. To disable server-side autocommit and enable client-side autocommit, set this option to YES (ClientAutocommit=yes). To restore the default behavior, set this option to NO (ClientAutocommit=no). This option is useful for frameworks that incur a tremendous performance penalty by constantly switching between manual commit and automatic commit.
Delphi	Delphi cannot handle multiple bookmark values for a row. When you set this value to NO, one bookmark value is assigned to each row, instead of the two that are otherwise assigned. Setting this option to YES can improve scrollable cursor performance.

Name	Description
DescribeCursor	<p>This parameter lets you specify how often you want a cursor to be re-described when a procedure is called. The default setting is If Required.</p> <p><b>Never</b></p> <p>Specify 0, N, or NO if you know that your cursors do not have to be re-described. Re-describing cursors is expensive and can decrease performance.</p> <p><b>If Required</b></p> <p>Specify 1, Y, or YES if you want the ODBC driver to determine whether a cursor must be re-described. ODBC does not describe the result set after a cursor is opened. This is the default setting.</p> <p><b>Always</b></p> <p>If you specify 2, A, or ALWAYS, the cursor is re-described each time it is opened. If you use Transact-SQL procedures or procedures that return multiple result sets, you must re-describe the cursor each time it is opened.</p>
Description	<p>This parameter allows you to provide a description of the ODBC data source.</p>
Driver	<p>This parameter allows you to specify an ODBC driver for the connection, as follows: <code>Driver=&lt;driver-name&gt;</code>. By default, the driver that is used is SAP IQ .</p> <p>The iqdsn Database Administration Utility automatically adds a Driver entry with the full path of the SAP IQ ODBC driver based on the setting of the LIBDIR16 environment variable.</p>
Escape	<p>This parameter specifies the escape character used in the LIKE clause of SQL statements that are generated by the ODBC driver when returning a list of tables or columns.</p> <p>By default the ODBC driver uses the tilde character (~), but some applications assume that the escape character is the backslash character (\).</p>
GetTypeInfoChar	<p>When this option is set to YES, CHAR columns are returned as SQL_CHAR instead of SQL_VARCHAR. By default, CHAR columns are returned as SQL_VARCHAR.</p>
InitString	<p>InitString allows you to specify a command that is run immediately after the connection is established. For example, you may want to set a database option or call a stored procedure.</p>

Name	Description
IsolationLevel	<p>You can specify one of the following values to set the initial isolation level for this data source:</p> <p><b>0</b></p> <p>This is also called the read uncommitted isolation level. This is the default isolation level. It provides the maximum level of concurrency, but dirty reads, non-repeatable reads, and phantom rows may be observed in result sets.</p> <p><b>1</b></p> <p>This is also called the read committed level. This provides less concurrency than level 0, but eliminates some of the inconsistencies in result sets at level 0. Non-repeatable rows and phantom rows may occur, but dirty reads are prevented.</p> <p><b>2</b></p> <p>This is also called the repeatable read level. Phantom rows may occur. Dirty reads and non-repeatable rows are prevented.</p> <p><b>3</b></p> <p>This is also called the serializable level. This provides the least concurrency, and is the strictest isolation level. Dirty reads, non-repeatable reads, and phantom rows are prevented.</p> <p><b>Snapshot</b></p> <p>You must enable snapshot isolation for the database to use this isolation level. The snapshot isolation levels prevent all interference between reads and writes. Writes can still interfere with each other. For contention, a few inconsistencies are possible and performance is the same as isolation level 0.</p> <p><b>Statement-snapshot</b></p> <p>You must enable snapshot isolation for the database to use this isolation level. The snapshot isolation levels prevent all interference between reads and writes. Writes can still interfere with each other. For contention, a few inconsistencies are possible and performance is the same as isolation level 0.</p> <p><b>Readonly-statement-snapshot</b></p> <p>You must enable snapshot isolation for the database to use this isolation level. The snapshot isolation levels prevent all interference between reads and writes. Writes can still interfere with each other. For contention,</p>

Name	Description
	a few inconsistencies are possible and performance is the same as isolation level 0.
KeysInSQLStatistics, MSApplications, MSApp	Set this parameter to YES if you want the SQLStatistics function to return foreign keys. The ODBC specification states that SQLStatistics should not return primary and foreign keys; however, some Microsoft applications (such as Microsoft Visual Basic and Microsoft Access) assume that primary and foreign keys are returned by SQLStatistics. MSApplications and MSApp are synonyms for KeysInSQLStatistics.
LazyAutocommit	Set this parameter to YES to delay the commit operation until a statement closes.
MatView	Normally, a materialized view is reported as a VIEW by schema functions such as SQLTables. Set this parameter to some string value if you want to distinguish materialized views from views. For example, MatView="MAT VIEW" causes the ODBC driver to report a materialized view as MAT VIEW.
MaxLength	Use this parameter to set SQL_ATTR_MAX_LENGTH. The default value is 262144 (256K) . Since LONG VARCHAR, LONG NVARCHAR, and LONG BINARY columns can range as large as 2147483647 bytes, 256K was chosen as a compromise for a default buffer length by the ODBC driver. This default value can be overridden by setting SQL_ATTR_MAX_LENGTH with the SQLSetConnectAttr or SQLSetStmtAttr functions. An ODBC application can also handle long column values using SQLGetData calls to fetch the data in chunks. However, there are interfaces such as System.Data.Odbc that do not provide the ability to access these features. The connection string "DSN=MyDataSource;DBN=demo;MaxLength=3145728" sets SQL_ATTR_MAX_LENGTH to 3 MB. The MaxLength parameter can also be set in a Data Source using the Microsoft ODBC Data Source Administrator (ODBCAD32) or iqdsn Database Administration Utility. In an ODBC application, the MaxLength parameter value, if it is specified, is overridden using SQL_ATTR_MAX_LENGTH with the SQLSetStmtAttr function.
PrefetchOnOpen	When PrefetchOnOpen is set to YES, a prefetch request is sent with a cursor open request. The prefetch eliminates a network request to fetch rows each time a cursor is opened. Columns must already be bound for the prefetch to occur on the open. This connection parameter can help reduce the number of client/server requests to help improve performance over a LAN or WAN.
PreventNotCapable	The SAP IQ ODBC driver returns an error because it does not support qualifiers. Some ODBC applications do not handle this error properly. Set this parameter to YES to prevent this error code from being returned, allowing these applications to work.

Name	Description
SkipRetryWithSingleRowMode	<p>Set this parameter to YES to prevent the ODBC driver from retrying a failed batch operation in single-row mode. For example, a batch INSERT of 100,000 rows could fail somewhere on a primary key violation. By default, the driver would then attempt single-row inserts up to the point of failure. This can be time-consuming. This option can be used to prevent this, realizing that the point of the failure will be more difficult to determine.</p>
SuppressInfoForDataTypes	<p>Set this parameter to a comma-separated list of data type names to prevent the ODBC driver from returning information about the specified data types. The specified data types are not returned by the SQLGetTypeInfo or the DatabaseMetaData.getTypeInfo functions. But, the application can use the data type in column declarations and in result set queries. For example, the following connection-string fragment prevents the ODBC driver from returning information about the NVARCHAR, LONG VARCHAR, and VARBIT data types.</p>
	<pre>SuppressInfoForDataTypes=nvarchar , long nvarchar , varbit</pre>
SuppressWarnings	<p>Set this parameter to YES to suppress warning messages that are returned from the database server on a fetch. Versions 8.0.0 and later of the database server return a wider range of fetch warnings than earlier versions of the software. For applications that are deployed with an earlier version of the software, you can select this option to ensure that fetch warnings are handled properly.</p>
TranslationDLL	<p>This option is provided for backward compatibility. The use of translators is not recommended.</p>
TranslationName	<p>This option is provided for backward compatibility. The use of translators is not recommended.</p>
TranslationOption	<p>This option is provided for backward compatibility. The use of translators is not recommended.</p>

# 21 iqheader Database Administration Utility

The `iqheader` utility is a dbspace header utility that determines which server, if any, is using a particular device, file, or LUN (Logical Unit Number) as a dbspace to analyze disk usage or to configure a multiplex query server.

## ⌘ Syntax

```
iqheader [ <dbspace_path> ]
```

## Parameters

### **dbspace\_path**

The device to be checked.

## Remarks

The `iqheader` utility reports the configuration of an arbitrary device regardless of whether it is currently in use by an SAP IQ server.

The user interface is a standalone console application called `iqheader` (`header.exe` on Windows). The `iqheader` tool searches the device for an SAP IQ dbspace header and reports the header information in a user-readable format.

## ⓘ Note

LUN is a logical unit number and is used to identify SCSI devices so the host can address and access the data on each disk drive in an array.

### **iqheader usage**

When invoked with no parameters, a usage summary is reported and a nonzero status is returned:

```
>iqheader
Usage: iqheader [dbspace_path]
```

### **iqheader error**

When the specified target is not a dbspace, an error message is reported and a nonzero status is returned:

```
>iqheader /dev/null
Not an IQ file: Error 0
```

### **Operating system errors**

When the specified target is unreadable, or any file operation fails due to an error returned from the operating system, the native operating system error appears and a nonzero status is returned:

```
>iqheader /dev/rdisk/c1t32d0s1 <
Open Failed: No such file or directory
>iqheader /dev/rdisk/c1t3d0s1 <
Open Failed: Permission denied
```

### iqheader output

When a valid dbspace is specified, `iqheader` prints the following dbspace configuration to the console and returns a 0 exit status:

- File Name – name of the file.
- Full Path – full path after symbolic link resolution
- Version – dbspace file format version
- File ID – unique number assigned to each dbspace
- Create Time – time of dbspace creation
- RW Mode – current read-write mode: RW, RO, RW, N/A (Upgraded)
- Last RW Mode – last dbspace mode
- Size (MB) – dbspace size, in megabytes
- Reserve (MB) – dbspace reserve size, in megabytes
- Block Size – size of block, in bytes
- Page Size – size of page, in bytes
- First Block – first block number mapped to dbspace
- Block Count – number of blocks that map to actual disk blocks
- Reserve Blocks – number of blocks that may be added to this dbspace
- Last Real Block – last block number that maps to an actual disk block
- Last Mapped Block – last block number mapped to dbspace
- OFlag – online status (YES/NO)
- Create ID – commit ID in which dbspace was created
- Alter ID – last commit ID in which dbspace was altered
- DBID1 – location of first database identity
- DBID2 – location of second database identity
- DBSpace ID – unique identifier assigned to each dbspace
- \_NextFLAllocLowerBank – lower bound of pre-allocate space for dbspace
- \_NextFLAllocUpperBank – upper bound of pre-allocate space for dbspace
- Pre-alter commit ID – commit\_id in the system tables ISYSDBFILE and ISYSIQDBSPACE
- \_ReqNumFreeListBlocks – number of blocks of type 'F' (free list blocks)

## Example

This example shows output for `iqheader`:

```
File Name: file1.iq
Full Path: /dev/dsk/file1.iq
DBFile Header Info
```

```
Version: 2
File ID: 16395
Create Time: 2008-06-02 21:57:00
RW Mode: RO
Last RW Mode: RW
Size (MB): 20
Reserve (MB): 20
Block Size: 8192
Page Size: 131072
First Block: 9408960
Block Count: 2560
Reserve Blocks: 2560
Last Real Block: 9411519
Last Mapped Block: 10454399
OFlags: 1
Create ID: 6905
Alter ID: 6964
DBID1: 0
DBID2: 0
DBSpace ID: 16395
_NextFLAllocLowerBank: 0
_NextFLAllocUpperBank: 0
Pre-alter commit ID: 6925
Dropped: NO
```

## 22 iqinit Database Administration Utility

The `iqinit` utility starts a database that gives the user executing the utility permission to create a database. The user must have access to the computer and file system.

### Syntax

```
iqinit [ <options> ] <new-database-file>
```

## Parameters

### options

#### **@data**

Reads in options from the specified environment variable or configuration file. If you want to protect passwords or other information in the configuration file, you can use the File Hiding utility to obfuscate the contents of the configuration file.

#### **-a**

Causes string comparisons to respect accent differences between letters (for example, `e` is less than `é` if the Unicode Collation Algorithm (UCA) is used for either CHAR or NCHAR data types (see `-z` and `-zn`). With the exception of Japanese databases created with a UCA collation, by default, accents are ignored (meaning `e` is equal to `é`). If all base letters (letters with accents and case removed) are otherwise equal, then accents are compared from left to right.

The default accent sensitivity of a UCA collation when creating a Japanese database is sensitive. That is, accents are respected.

#### **-af**

Causes string comparisons to respect accent differences between letters (for example, `e` is less than `é`) if the UCA is used for either CHAR or NCHAR data types (see `-z` and `-zn`). By default, accents are ignored (meaning `e` is equal to `é`). If all base letters (letters with accents removed) are otherwise equal, then accents are compared from right to left, consistent with the rules of the French language.

#### **-b**

Blank pads the database.

The database server compares all strings as if they are varying length and stored using the VARCHAR domain. This includes string comparisons involving fixed length CHAR or NCHAR columns. In addition, SAP SQL Anywhere never trims or pads values with trailing blanks when the values are stored in the database.

By default, the database server treats blanks as significant characters. So, the value `'a '` (the character `'a'` followed by a blank) is not equivalent to the single-character string `'a'`. Inequality comparisons also treat a blank as any other character in the collation.

If blank padding is enabled (the `iqinit -b` option), the semantics of string comparisons more closely follow the ANSI/ISO SQL standard. With blank-padding enabled, SAP SQL Anywhere ignores trailing blanks in any comparison.

In the example above, an equality comparison of 'a ' to 'a' in a blank-padded database returns TRUE. With a blank-padded database, fixed-length string values are padded with blanks when they are fetched by an application. The `ansi_blanks` connection option controls whether the application receives a string truncation warning on such an assignment.

#### **-c**

Considers all values case sensitive in comparisons and string operations. Identifiers in the database are case insensitive, even in case sensitive databases.

With the exception of Japanese databases created with a UCA collation, the default behavior is that all comparisons are case insensitive. The default case sensitivity of a UCA collation when creating a Japanese database is sensitive.

This option is provided for compatibility with the ISO/ANSI SQL standard.

#### **-dba [ DBA-user ] [ pwd ]**

Specifies the DBA user ID and password. If you specify a new name for the DBA user for the database, you can no longer connect to the database as the user DBA. You can also specify a different password for the DBA database user. If you do not specify a password, the default password `sql` is used. If you do not specify this option, the default user ID DBA with password `sql` is created.

The following command creates a database with a DBA user named `testuser` with the default password `sql`:

```
iqinit -dba testuser, mydb.db
```

The following command uses the default user ID DBA with password `mypwd`:

```
iqinit -dba ,mypwd mydb.db
```

The following command changes the DBA user to `user1` with password `mypwd`:

```
iqinit -dba user1,mypwd mydb.db
```

#### **-dbs size[ k | m | g | p ]**

Pre-allocates space for the database. Pre-allocating space for the database helps reduce the risk of running out of space on the drive the database is located on. As well, it can help improve performance by increasing the amount of data that can be stored in the database before the database server needs to grow the database, which can be a time-consuming operation.

By default, the size is in bytes. You can use `k`, `m`, `g`, or `p` to specify units of kilobytes, megabytes, or gigabytes, or pages, respectively.

#### **-ea algorithm**

Specifies the encryption algorithm used for database or table encryption (`-et`). Specify `-ea simple` for simple encryption (do not specify `-ek` or `-ep`). Simple encryption is equivalent to obfuscation and is intended only to keep data hidden in the event of casual direct access of the database file, to make it more difficult for someone to decipher the data in your database using a disk utility to look at the file. Simple encryption is not secure.

For greater security, specify one of AES, AES256, AES256CTR, or any of the FIPS-certified variants (for example, AES\_FIPS, AES256CTR\_FIPS). AES and AES\_FIPS use 128-bit strong encryption while all the others are 256-bit strong. AES, AES256, and their FIPS variants encrypt with CBC block cipher mode. AES256CTR and AES256CTR\_FIPS encrypt with CTR block cipher mode.

Specify ARIA256 or ARIA256CTR for ARIA 256-bit encryption with CBC block cipher mode or CTR block cipher mode, respectively. ARIA is a variant of AES and considered of equivalent strength.

For encryption aliases, see [Encryption Algorithm Aliases](#).

For strong encryption, you must also specify the `-ek` or `-ep` option.

To create a database that is not encrypted, specify `-ea none`, or do not include the `-ea` option (and do not specify `-et`, `-ep`, or `-ek`). If you do not specify the `-ea` option, the default behavior is as follows:

- `-ea none`, if `-ek`, `-ep`, or `-et` is not specified
- `-ea AES`, if `-ek` or `-ep` is specified (with or without `-et`)
- `-ea simple`, if `-et` is used without `-ek` or `-ep`

Algorithm names are case-insensitive.

File compression utilities cannot compress encrypted database files as much as unencrypted ones.

#### **-ek key**

Specifies that you want to create a strongly encrypted database by specifying an encryption key directly in the command. The `-ek` option is used with an AES algorithm, optionally specified using the `-ea` option. If you specify the `-ek` option without specifying the `-ea` option, AES is used by default. When specified with `-et`, the database is not encrypted. Instead, table encryption is enabled.

#### **-ep**

Specifies that you want to create a strongly encrypted database by inputting the encryption key in a window. This provides an extra measure of security by never allowing the encryption key to be seen in clear text. You must input the encryption key twice to confirm that it was entered correctly. If the keys don't match, the initialization fails. When specified with `-et`, the database is not encrypted. Instead, table encryption is enabled.

#### **-et**

Enables table encryption using the encryption algorithm (and key) specified for the `-ea` option. Use this option when you want to create encrypted tables instead of encrypting the entire database. If you specify `-et` with `-ek` or `-ep`, but not `-ea`, the AES algorithm is used by default. When you specify only `-et`, simple encryption is used.

Enabling table encryption does not mean your tables are encrypted. You must encrypt tables individually, after database creation.

When table encryption is enabled, table pages for the encrypted table, associated index pages, and temporary file pages are encrypted, and the transaction log pages that contain transactions on encrypted tables.

The following example creates the database `new.db` with strong encryption enabled for tables using the key `abc`, and the AES\_FIPS 128-bit encryption algorithm:

```
iqinit -et -ek abc -ea AES_FIPS new.db
```

#### **-i**

Excludes jConnect for JDBC system objects from the database. To use the jConnect JDBC driver to access system catalog information, you need jConnect catalog support (installed by default). When you specify this option, you can still use JDBC, as long as you do not access system information. You can add jConnect support at a later time using the `ALTER DATABASE` statement.

**-iqblksize**

The I/O transfer block size in bytes.

**-iqmsgpath**

The path name of the segment containing the SAP IQ message trace file.

**-iqpath**

The path name of the main segment file containing the SAP IQ data.

**-iqpgsize**

The page size in bytes for the SAP IQ segment of the database.

**-iqreservesize**

The size, in MB, of the space to reserve for the IQ main store.

**-iqsize**

The size, in MB, of either the raw partition or OS file with the `-iqpath`, in the `IQ_SYSTEM_MAIN` dbspace.

**-iqtmppath**

The path name of the temporary segment file.

**-iqtmpreservesize**

The size, in MB, to reserve for the temporary IQ store.

**-iqtmpsize**

The size, in MB, of either the raw partition or OS file for the `-iqtmppath`, in the `IQ_SYSTEM_TEMP` dbspace.

**-k**

Does not create the `SYSCOLUMNS` and `SYSINDEXES` views. By default, database creation generates the views `SYS.SYSCOLUMNS` and `SYS.SYSINDEXES` for compatibility with system tables that were available in Watcom SQL (versions 4 and earlier of this software). These views conflict with the SAP ASE compatibility views `dbo.syscolumns` and `dbo.sysindexes`.

**-kdi number**

Specifies the number of times that the encryption key is hashed. Specify a whole number between 1 and 1000. The default value is 2, which results in the encryption key being hashed 2000 times. The higher the number, the better security. If you are running the database on a slow computer, then a very high number could result in the database taking longer to start. Use the `-li` option to estimate how this setting could affect the startup time (once the database is running, there is no performance impact). The `-kdi` option must be specified with either the `-ek` or the `-ep` option.

**-l**

Lists the available collation sequences and then stops. No database is created. To specify a collation sequence, use the `-z` option.

**-le**

Lists the available character set encodings and then stops. No database is created. Each character set encoding is identified by one or more labels. These are strings that can be used to identify the

encoding. Each line of text that appears lists the encoding label and alternate labels by which the encoding can be identified. These labels fall into one of several common categories: SA (the SAP SQL Anywhere label), IANA (Internet Assigned Numbers Authority), MIME (Multipurpose Internet Mail Extensions), ICU (International Components for Unicode), JAVA, or ASE (Adaptive Server Enterprise).

If you want to view a list of character set encodings that includes the alternate labels, specify the `-le+` option.

When the `iqinit` utility reports the character set encoding, it always reports the database server version of the label. For example, the following command reports the CHAR character set encoding `windows-1250`:

```
iqinit -ze cp1250 -z uca test.db
```

**-m filename**

Creates a transaction log mirror. A transaction log mirror is an identical copy of a transaction log, usually maintained on a separate device, for greater protection of your data. By default, the database server does not use a transaction log mirror.

**-n**

Creates a database without a transaction log. Creating a database without a transaction log saves disk space, but can result in poorer performance because each commit causes a checkpoint. Also, if your database becomes corrupted and you are not running with a transaction log, data is not recoverable. The transaction log is required for data replication and provides extra security for database information during a media or system failure.

**-o filename**

Writes output messages to the named file.

**-p page-size**

Specifies the page size for the database. The page size for a database can be (in bytes) 2048, 4096, 8192, 16384, or 32768, with 4096 being the default. Use `k` to specify units of kilobytes (for example, `-p 4k`).

Large databases can benefit from a larger page size. For example, the number of I/O operations required to scan a table is generally lower, as a whole page is read in at a time. However, there are additional memory requirements for large page sizes. It is strongly recommended that you do performance testing (and testing in general) when choosing a page size. Then choose the smallest page size that gives satisfactory results. For most applications, 16 KB or 32 KB page sizes are not recommended. You should not use page sizes of 16 KB or 32 KB in production systems unless you can be sure that a large database server cache is always available, and only after you have investigated the trade offs of memory and disk space with its performance characteristics. If a large number of databases are going to be started on the same server, pick a reasonable page size.

**-pd**

If this option is not specified (default), all privileged system procedures execute with the privileges of the invoker. If specified, pre-16.0 privileged system procedures execute with the privileges of the definer, while 16.0 or later privileged system procedures execute with the privileges of the invoker.

**-q**

Runs in quiet mode — messages are not displayed.

**-s**

Adds global checksums (a checksum is added to each database page). By default, this option is on. Checksums are used to determine whether a database page has been modified on disk. When you create a database with global checksums enabled, a checksum is calculated for each page just before it is written to disk. The next time the page is read from disk, the page's checksum is recalculated and compared to the checksum stored on the page. If the checksums are different, then the page has been modified or corrupted on disk, and an error occurs. Critical database pages are always checksummed by the database server, regardless of the value of the `-s` option.

Checksums are automatically enabled for databases running on storage devices such as removable drives to help provide early detection if the database becomes corrupt.

If a database is created with global checksums disabled, you can still add checksums to pages when they are written by using the `-wc` option or the `START DATABASE` statement.

#### **-t transaction-log-name**

Specifies the name of the transaction log file. The transaction log is a file where the database server logs all changes, made by all users, no matter what application is being used. The transaction log plays a key role in backup and recovery, and in data replication. If the file name has no path, it is placed in the same directory as the database file. If you run `iqinit` without specifying `-t` or `-n`, a transaction log is created with the same file name as the database file, but with extension `.log`.

#### **-z coll [ collation-tailoring-string**

Specifies the collation sequence for the database. The collation sequence is used for sorting and comparing character data types (CHAR, VARCHAR, and LONG VARCHAR). The collation provides character comparison and ordering information for the encoding (character set) being used. It is important to choose your collation carefully. It cannot be changed after the database has been created without unloading and reloading the database. If the collation is not specified, the database server chooses a collation based on the operating system language and character set. To view the available collation sequences, see the `-l` option.

Optionally, you can specify collation tailoring options (`<collation-tailoring-string>`) for additional control over the sorting and comparing of characters. These options take the form of keyword=value pairs, assembled in parentheses, following the collation name. For example:

Case and accent settings specified in the `<collation-tailoring-string>` override case and accent options for `iqinit` (`-c`, `-a`, and `-af`), if you specify both.

#### **-ze encoding**

Specifies the encoding for the collation. Most collations specified by `-z` dictate both the encoding (character set) and ordering. For those collations, `-ze` should not be specified.

If the collation specified by `-z` is Unicode Collation Algorithm (UCA), then `-ze` can specify UTF-8 or any single-byte encoding for CHAR data types.

By default, the database server uses UTF-8. Use `-ze` to specify a locale-specific encoding and get the benefits of the UCA for comparison and ordering.

#### **-zn coll [ collation-tailoring-string**

Specifies the collation sequence used for sorting and comparing of national character data types (NCHAR, NVARCHAR, and LONG NVARCHAR). The collation provides character ordering information for the UTF-8 encoding (character set) being used. Values are UCA (the default), or UTF8BIN which provides a binary ordering of all characters whose encoding is greater than 0x7E. If the `dbicu16` and `dbicudt16` DLLs are not installed, then the default NCHAR collation is UTF8BIN

Optionally, you can specify collation tailoring options (`<collation-tailoring-string>`) for additional control over the sorting and comparing of characters. These options take the form of keyword=value pairs, assembled in parentheses, following the collation name. For example:

```
iqinit -c -zn UCA(case=LowerFirst) sens.db
```

Case and accent settings specified in the `<collation-tailoring-string>` override case and accent options for `iqinit` (`-c`, `-a`, and `-af`), if you specify both.

#### **new-database-file**

The new database file name.

## Remarks

`iqinit` lets you create an SAP IQ database from the command line without starting a database:

- If no `iqinit` command line parameters are specified, `iqinit` creates an SAP SQL Anywhere database.
- If the `-iqpath` command line parameter is specified, `iqinit` creates an SAP IQ database.


## Example

This command creates a SAP IQ database called `bar.iq`:

```
$ iqinit -iqpath bar.iq -iqsize 20M
-iqpgsize 2048
-iqreservesize 10M bar.db
SQL Anywhere Initialization Utility Version 16.0.0.5530
Debug
CHAR collation sequence:
ISO_BINENG(CaseSensitivity=Respect)
CHAR character set encoding: ISO_8859-1:1987
NCHAR collation sequence:
UCA(CaseSensitivity=UpperFirst;
AccentSensitivity=Respect;
PunctuationSensitivity=Primary)
NCHAR character set encoding: UTF-8
Creating system tables
Creating system views
Setting option values
Database "bar.db" created successfully
```

## 23 iqocscfg Configuration Utility

The `iqocscfg` utility is a Windows-only Open Client and Open Server configuration utility you can use to configure environment variables, directory drivers, and security drivers.

See [Open Server 15.5 > Software Developer's Kit 15.5 > Open Client and Open Server Configuration Guide for Microsoft Windows > Using ocscfg](#) .

## 24 iqsqlpp SQL Preprocessor Utility

The SAP IQ SQL preprocessor utility `iqsqlpp` translates the SQL statements in an input file (`.sql`) into C language source that is put into an output file (`.c`).

### Syntax

```
iqsqlpp [ <options> ] <in filename> [ <output-filename> ]
```

### Parameters

#### options

**-d**

Favor data size.

**-e level**

Flag nonconforming SQL syntax as an error.

The allowed values of `<level>` are:

- `c03` – flag syntax that is not core SQL/2003 syntax
- `p03` – flag syntax that is not full SQL/2003 syntax
- `c99` – flag syntax that is not core SQL/1999 syntax
- `p99` – flag syntax that is not full SQL/1999 syntax
- `e92` – flag syntax that is not entry-level SQL/1992 syntax
- `i92` – flag syntax that is not intermediate-level SQL/1992 syntax
- `f92` – flag syntax that is not full-SQL/1992 syntax
- `t` – flag syntax that is not full-SQL/1992 syntax
- `u` – flag non-standard host variable types

The following are also supported for compatibility with previous versions:

- `e` – for entry-level SQL92
- `i` – for intermediate-level SQL92
- `f` – for full SQL92
- `w` – to allow all supported syntax

**-h width**

Limit the maximum line length of output.

**-k**

Include user declaration of `SQLCODE`.

**-n**

Line numbers.

**-o O/S spec**

Target operating system specification (WINDOWS, WINNT or UNIX).

**-q**

Quiet mode — do not print banner.

**-r**

Generate reentrant code.

**-s len**

Maximum string constant length for the compiler.

**-w level**

Flag nonconforming SQL syntax as a warning.

The allowed values of `<level>` are:

- `c03` – flag syntax that is not core SQL/2003 syntax
- `p03` – flag syntax that is not full SQL/2003 syntax
- `c99` – flag syntax that is not core SQL/1999 syntax
- `p99` – flag syntax that is not full SQL/1999 syntax
- `e92` – flag syntax that is not entry-level SQL/1992 syntax
- `i92` – flag syntax that is not intermediate-level SQL/1992 syntax
- `f92` – flag syntax that is not full-SQL/1992 syntax
- `t` – flag syntax that is not full-SQL/1992 syntax
- `u` – flag non-standard host variable types

**-x**

Change multibyte SQL strings to escape sequences.

**-z cs**

Specify the collation sequence. For a list of recommended collation sequences, enter `iqinit -l` at a command prompt.

**in filename** The input file.

**output-filename** The output file.

## Remarks

Embedded SQL is a database programming interface for the C and C++ programming languages. Embedded SQL consists of SQL statements intermixed with (embedded in) C or C++ source code. These SQL statements are translated by an SQL preprocessor into C or C++ source code, which you then compile.

The SQL preprocessor processes a C or C++ program containing Embedded SQL before the compiler is run. `iqsqlpp` translates the SQL statements in the input file `<sql-filename>` into C language source that is put into the `<output-filename>`. The normal extension for source programs with Embedded SQL is `.sqlc`. The default output file name is the `<sql-filename>` with an extension of `.c`. If the `<sql-filename>` has a `.c` extension, the default output file name extension is `.cc`.

## 25 iqlsunload Local Store Unload Utility

A utility that you can use to unload the contents of an IQ 12.7 local store, and unload schema and data from user tables in an IQ 12.7 local store.

In current multiplex configurations, multiple nodes can write to the main store, which eliminates the need for local stores. `iqlsunload` is a command line utility that you can use to unload a 12.7 local store. `iqlsunload` is used only in 12.7 ESD #5 multiplex migrations.

To use `iqlsunload` to unload the contents of an SAP IQ 12.7 database, see *iqunload for Local Store Unload* in the installation documentation for your platform.

### Related Information

[iqlsunload for Local Store Unload](#)

# 26 iqunload Schema and Data Export Utility

A command line utility to unload a full or partial schema based on a subset of tables and users. Data can optionally be included in the unload.

Run `iqunload` against an SAP IQ 15.4 or later database to unload the database schema, and optionally, its data into a SQL script file. You can then load the schema file into another SAP IQ database of the same version.

## Note

When you run `iqunload` against an SAP IQ 12.7 database, `iqunload` instead migrates the database, re-creating the legacy catalog on the new database catalog in the current installation. To use `iqunload` to migrate a SAP IQ 12.7 database, see *iqunload Utility* in the *SAP IQ Installation and Update Guide* documentation for your platform.

See the *SAP IQ Administration: Database* documentation for details on exporting schema.

## Syntax

```
iqunload [ <parameters> ] <directory> [ @<data> ]
```

## Parameters

### parameters

#### **-c "keyword = value, ..."**

Supplies database connection parameters. Use the DBF parameter to specify the name of the database file for unload. The file path is either absolute or relative to the server startup directory.

For example, the following statement unloads the database `mydemo`, connecting as user `DBA` with password `sql`. The data is unloaded into the `c:\unload` directory:

```
iqunload -c "DBF=c:\IQdemo\mydemo.db;UID=DBA;PWD=<password>" c:\unload
```

#### **-dc**

(Optional) forces all computed columns in the database to be recalculated.

#### **-n**

Unloads the schema definition only; no data.

#### **-null\_string text**

(Optional) applies to tables containing a `varchar` or `varbinary` data types only. During unload, provides a means to differentiate between `NULL` values and string values that contain the word `NULL`. The value of `-null_string` should not match any other pattern of values in the table.

If not defined, defaults to `NULL`.

#### **-o messages\_file\_name**

(Optional) logs output messages to the named file. The location of this file is relative to `iqunload`.

**-q**

(Optional) suppresses messages and windows.

#### Note

When using `-q`, also specify `-y` or `iqunload` fails if the generated SQL script file already exists.

**-r script\_file\_name**

(Optional) specifies the name and directory of the generated SQL script file. The default is `reload.sql` in the current directory.

**-t table\_list**

(Optional) outputs listed tables only. Identify tables in a comma separate list as `<OwnerName>.<TableName>` or `<TableName>` alone.

#### Note

Cannot be used with `-t1` parameter.

**-t1 'table\_pattern1,table\_pattern2,...'**

(Optional) unloads tables matching the specified pattern, listed in a comma separate list.

When used in conjunction with `-u1`, if a specified table is owned by a user not included in the `-u1` list, the table is not unloaded.

When used in conjunction with `-t1d`, if the same table is not specified by both clauses, only the schema for the specified table is unloaded. For example, `-t1 table1, table2 -t1d table1` unloads the schema and data for `table1`, but the schema only for `table2`.

Cannot be used with `-t` parameter.

When `-n` not specified, all data for specified tables is unloaded.

**-t1d 'table\_pattern1:condition1, table\_pattern2:condition2'**

(Optional) restricts the unloaded of data matching the specified pattern within tables. List tables with conditions in a comma separated list.

To further restrict the data unload within the table, use a colon (:) followed by a `where` clause. All data in the specified table is unloaded if no `where` clause is specified.

When used with `-t1`, if the same table is not specified by both clauses, no data is unloaded for the specified table. For example, `-t1 table1 -t1d table1, table2` unloads the schema and data for `table1`, but no schema or data for `table2`.

When `-t1` not specified, all tables are considered for unload, then conditions apply, but no schema or data for

When `-n` is also specified, no data is unloaded regardless of conditions.

**-ul 'user\_pattern1, userpattern2, ...'**

(Optional) unloads objects owned by users matching the specified comma separate list.

**-up**

(Optional) unloads user passwords (hashed value).

**-v**

(Optional) displays the name of the table being unloaded, and the number of rows that have been unloaded.

**-y**

(Optional) replaces existing schema SQL script with new output without confirmation.

#### ⓘ Note

When using `-q`, also specify `-y` or `iqunload` fails if the generated SQL script file already exists.

#### **directory**

Specifies the directory where the data files referenced in the unloaded

#### ⓘ Note

Not required when used with `-n`.

#### **@data**

Reads in options from the specified environment variable or configuration file.

## Remarks

The generated schema must be loaded into a database of the same version as the database from which was unloaded.

The `-t1`, `-t1d`, and `-u1` parameters support pattern matching to define a subset of tables and users for unload. The pattern list is read iteratively from left to right, building a temporary list of objects to unload. If the first character of a pattern is not an apostrophe (!), all objects matching the pattern are added to the temporary list. If the first character of the pattern is an apostrophe, any object on the temporary list matching the pattern is removed from the list.

For example, when building the temporary list, the syntax `-t1 '!table1,table1'` would first remove all tables named `table1` from the temporary list and then re-add all tables named `table1`. The net result is only tables named `table1` are unloaded. Conversely, the syntax `-t1 'table1,!table1'` first limits the temporary list to only tables named `table1` and then removes all tables named `table1` from the list. The net result now is that no tables are unloaded.

Pattern-matching uses SQL `LIKE` syntax (`%`, `*`, `?`, etc.)

If no `-u1` parameter is specified, there are no restrictions on owners for tables unloaded.

When `-n` is specified, no data unloaded regardless of other clauses. When `-n` is not specified, the data unloaded is dependent on other clauses specified.

To exclude users or tables in an `IQUNLOAD` statement, you must first specify the pool of users or tables to be considered for the exclusion. For example, to exclude tables beginning with the letter `t` (`!t%`), you must first include all tables (`%`): `-t1 '%,!t%'`.

## Permissions

Requires DBA authority for 15.4 databases.

Requires the SELECT ANY TABLE system privilege for 16.0 and later databases.

## Examples

- This example exports the database schema, and exports encrypted passwords:

```
iqunload -n -c "uid=DBA;pwd=*****;eng=iq_iqdemo;dbn=iqdemo;" -up -r
iqdemo_unload.sql
```

### Note

If you do not specify the `-up` option, then all `GRANT CONNECT TO <user>` commands are printed without the `IDENTIFY BY` clause in the generated file. This means users do not have a password, and cannot connect to the database.

- This example unloads all tables beginning with the letter `t`:

```
iqunload -n -c "... " -tl 't%'
```

- This example unloads all tables not beginning with the letter `t`:

```
iqunload -n -c "... " -tl '%,!t%'
```

- This example unloads all table named `table1`, or table owned by user `user1` that begins with the letter `t`, excluding any table named `t2`:

```
iqunload -n -c "... " -tl 'table1,user1.t%,!t2'
```

- This example unloads any table beginning with the letters `ta` and owned by a user name beginning with `sm`:

```
iqunload -n -c "... " -tl 'sm%.ta%'
```

- This example unloads any table owned by a user name containing the letters `em` but not equal to `temp`:

```
iqunload -n -c "... " -ul '%em%,!temp'
```

If tables `u1.table1` and `u2.table1` exist, both tables are unloaded by this syntax:

```
iqunload -n -c "... " -tl 'table1'
```

But only `u1.table1` is unloaded if the owner is included in the syntax:

```
iqunload -n -c "... " -tl 'u1.table1'
```

- This example unloads only rows for `table1` where column `c_char1` is not null. For `table2`, only rows where `vb` is null are unloaded. For all other tables (`%`), all data is unloaded. The data is unload to the directory `./data` relative to the server, in a file named `data_reload.sql`:

```
iqunload -c "... " -tld '%,table1:c_char1 is not null, table2:vb is null'-r
data_reload.sql ./data
```

- This example unloads schema for all tables of the form `ta%`. It unloads all data for tables `table1`, but only those rows in `table6` where `c1 < 0`. The data is unload to the directory `./data` relative to the server, in a file named `data_reload.sql`:

```
iqunload -c "... " -tl 'ta%' -tld '%.ta%:none, table1, table6:c1 < 0, !table3'
-r data_reload.sql ./data
```

- This example unloads schema and data for `table1` and uses `NULL_NULL` for `null_string`. The data is unload to the directory `./data` relative to the server, in a file named `data_reload.sql`.

```
iqunload -c "... " -tl 'table1' -null_string NULL_NULL -r data_reload.sql ./
data
```

## Related Information

[SAP IQ 12.7 Upgrade and Database Migration](#)  
[iqunload for SAP IQ 12.7 Database Migration](#)

## 27 isql Interactive SQL Utility

The `isql` utility is a command line Interactive SQL utility that uses the Open Client API.

### ↔ Syntax

```
isql
[ -b ] [ -e ] [ -F ] [ -p ] [ -n ] [ -v ] [ -W ] [ -X ] [ -Y ] [ -Q ]
  [ -a <display_charset> ]
  [ -A <packet_size> ]
  [ -c <cmdend> ]
  [ -D <database> ]
  [ -E <editor> ]
  [ -h <header> ]
  [ -H <hostname> ]
  [ -i <inputfile> ]
  [ -I <interfaces_file> ]
  [ -J <client_charset> ]
  [ -K <keytab_file> ]
  [ -l <login_timeout> ]
  [ -m <errorlevel> ]
  [ -o <outputfile> ]
  [ -P <password> ]
  [ -R <remote_server_principal> ]
  [ -s <colseparator> ]
  [ -S <server_name> ]
  [ -t <timeout> ]
-U <username>
[ -V [ <security_options> ] ]
[ -w <columnwidth> ]
[ -z <locale_name> ]
[ -Z <security_mechanism> ]
[ --conceal ]
[ --URP ]
```

### Parameters

**-b**

Disables display of the table headers output.

**-e**

Echoes input.

**-F**

Enables the FIPS flagger. When you specify the `-F` parameter, the server returns a message when it encounters a non-standard SQL command. This option does not disable SQL extensions. Processing completes when you issue the non-ANSI SQL command.

**-p**

Prints performance statistics.

**-n**

Removes numbering and the prompt symbol (>) from the echoed input lines in the output file when used in conjunction with `-e`.

**-v**

Prints the version number and copyright message for `isql` and then exits.

**-X**

Initiates the login connection to the server with client-side password encryption. `-x` enables both extended password encrypted connections and password encrypted connections without plain text password reconnection. `isql` (the client) specifies to the server that password encryption is desired. The server sends back an encryption key, which `isql` uses to encrypt your password, and the server uses the key to authenticate your password when it arrives.

If `isql` crashes, the system creates a core file that contains your password. If you did not use the encryption option, the password appears in plain text in the file. If you used the encryption option, your password is not readable

**-W**

Disables both extended password and password encrypted negotiations.

**-Y**

Tells the Adaptive Server to use chained transactions.

**-Q**

Provides clients with failover property.

**-a display\_charset**

Runs `isql` from a terminal whose character set differs from that of the machine on `isql` is running. Use `-a` with `-J` to specify the character set translation file (`.xlt` file) required for the conversion. Use `-a` without `-J` only if the client character set is the same as the default character set.

### Note

The `ascii_7` character set is compatible with all character sets. If either the Adaptive Server character set or the client character set is set to `ascii_7`, any 7-bit ASCII character can pass unaltered between client and server. Other characters produce conversion errors.

**-A packet\_size**

Specifies the network packet size to use for this `isql` session. For example, the following sets the packet size to 4096 bytes for this `isql` session:

```
isql -A 4096
```

- To check your network packet size, enter:

```
SELECT * FROM sysprocesses
```

- The value is displayed under the `network_pktsize` heading.
- `<size>` must be between the values of the default `network packet size` and `maximum network packet size configuration` parameters, and must be a multiple of 512.
- Use larger-than-default packet sizes to perform I/O-intensive operations, such as `readtext` or `writetext` operations.
- Setting or changing Adaptive Server's packet size does not affect the packet size of remote procedure calls.

**-c cmdend**

Changes the command terminator. By default, you terminate commands and send them to by typing "go" on a line by itself. When you change the command terminator, do not use SQL reserved words or control characters.

**-D database**

Selects the database in which the `isql` session begins.

**-E editor**

Specifies an editor other than the default editor, `vi`. To invoke the editor, enter its name as the first word of a line in `isql`.

**-h headers**

Specifies the number of rows to print between column headings. The default prints headings only once for each set of query results.

**-H hostname**

Specifies the number of rows to print between column headings. The default prints headings only once for each set of query results.

**-i interfaces\_file**

Specifies the name and location of the interfaces file to search when connecting to Adaptive Server. If you do not specify `-i`, `isql` looks for a file named `interfaces` in the directory specified by your IQ environment variable.

**-J client\_charset**

Specifies the character set to use on the client. `-J <client_charset>` requests that Adaptive Server convert to and from `<client_charset>`, the character set used on the client. A filter converts input between `<client_charset>` and the Adaptive Server character set.

`-J` with no argument sets character set conversion to NULL. No conversion takes place. Use this if the client and server use the same character set.

Omitting `-J` sets the character set to a default for the platform. The default may not necessarily be the character set that the client is using. For more information about character sets and the associated flags, see *Configuring Client/Server Character Set Conversions* in [SAP Adaptive Server Enterprise System Administration Guide: Volume 1](#).

**-K keytab\_file**

Specifies the path to the keytab file used for authentication in DCE.

**-l login\_timeout**

Specifies the maximum timeout value allowed when connecting to Adaptive Server. The default is 60 seconds. This value affects only the time that `isql` waits for the server to respond to a login attempt. To specify a timeout period for command processing, use the `-t <timeout>` parameter.

**-m errorlevel**

Customizes the error message display. For errors of the severity level specified or higher, only the message number, state, and error level are displayed; no error text appears. For error levels lower than the specified level, nothing appears.

**-o outputfile**

Specifies the name of an operating system file to store the output from `isql`. Specifying the parameter as `-o <outputfile>` is similar to `> outputfile`.

**-P password**

Specifies your Adaptive Server password. If you do not specify the `-P` flag, `isql` prompts for a password. If your password is NULL, use the `-P` flag without any password.

**-R remote\_server\_principal**

Specifies the principal name for the server as defined to the security mechanism. By default, a server's principal name matches the server's network name (which is specified with the `-s` parameter or the `DSQUERY` environment variable). Use the `-R` parameter when the server's principal name and network name are not the same.

**-s colseparator**

Resets the column separator character, which is blank by default. To use characters that have special meaning to the operating system (for example, "|", ";", "&", "<", ">"), enclose them in quotes or precede them with a backslash.

**-S server\_name**

Specifies the name of the Adaptive Server to which to connect. `isql` looks this name up in the interfaces file. If you specify `-s` with no argument, `isql` looks for a server named IQ. If you do not specify `-s`, `isql` looks for the server specified by your `DSQUERY` environment variable.

**-t timeout**

Specifies the number of seconds before a SQL command times out. If you do not specify a timeout, the command runs indefinitely. This affects commands issued from within `isql`, not the connection time. The default timeout for logging into `isql` is 60 seconds.

**-U username**

Specifies a login name. Login names are case sensitive.

**--URP string**

Allows you to supply a login redirection string for an SAP IQ server. This example specifies the logical server, database, node type and redirection setting:

```
isql -Usa -P --URP "LS=salogsrv;dbname=dbone;
node=writer;redirect=no"
```

Alternatively, it enables the setting of the universal remote password `<remotepassword>` for clients accessing Adaptive Server. The application sets the universal remote password. For example, `ctlib` uses `ct_remote_pwd()` and `jdbc` uses the `setRemotePassword` method.

**-V security\_options**

Specifies network-based user authentication. With this option, the user must log in to the network's security system before running the utility. In this case, users must supply their network user name with the `-U` option; any password supplied with the `-P` option is ignored.

You can follow `-V` with a `<security_options>` string of key-letter options to enable additional security services. These key letters are:

- `-c` – enable data confidentiality service
- `-i` – enable data integrity service
- `-m` – enable mutual authentication for connection establishment
- `-o` – enable data origin stamping service
- `-q` – enable out-of-sequence detection
- `-r` – enable data replay detection

### **-w columnwidth**

Sets the screen width for output. The default is 80 characters. When an output line reaches its maximum screen width, it breaks into multiple lines.

### **-z locale\_name**

Specifies the official name of an alternate language to display isql prompts and messages. Without `-z`, `isql` uses the server's default language. You can add languages to an Adaptive Server during installation or afterward, using the `langinstall` utility (`langinst` in Windows) or the `sp_addlanguage` stored procedure.

### **-Z security\_mechanism**

Specifies the name of a security mechanism to use on the connection.

Security mechanism names are defined in the `libtcl.cfg` configuration file located in the `ini` subdirectory below the IQ installation directory. If no `<security_mechanism name>` is supplied, the default mechanism is used. For more information on security mechanism names, see the description of the `libtcl.cfg` file in the *Open Client and Open Server Configuration Guide*.

### **--conceal**

Hides your input during an `isql` session. The `--conceal` option is useful when entering sensitive information, such as passwords. The syntax for the `--conceal` option is:

```
isql --conceal [':?' | 'wildcard']
```

The maximum length of `<wildcard>`, a 32-byte variable, is 80 characters.

#### **Note**

`--conceal` is silently ignored in batch mode.

## Remarks

You cannot create user-defined database options in `isql`. If you need to add your own database options, use the `dbisql` Interactive SQL utility instead.

Use this syntax for `isql_r` if you are using threaded drivers.

You must set the IQ environment variable to the location of the current version of Adaptive Server before you can use `isql`.

This following lists the commands you can execute within interactive `isql`:

#### **:r filename**

Reads an operating system file into the command buffer.

Do not include the command terminator in the file; once you have finished editing, enter the terminator interactively on a line by itself.

#### **:R filename**

Reads an operating system file into the command buffer and then displays it.

Do not include the command terminator in the file; once you have finished editing, enter the terminator interactively on a line by itself.

### **use database\_name**

Changes the current database.

### **!! os\_command**

Executes an operating system command. Place at the start of a line.

### **> file\_name**

Redirects the output of the T-SQL command to `<file_name>`. The following example inserts the server version into `<file_name>`:

```
select @@version
go > file_name
```

### **>> file\_name**

Appends the output of the T-SQL command to `<file_name>`. The following example appends the server version into `<file_name>`:

```
select @@version
go >> file_name
```

### **| command**

Pipes the output of the T-SQL command to an external command.

The following example finds all instances of "sa" in the listing produced by `sp_who`:

```
sp_who
go | grep sa
```

### **vi (Linux), edit (Windows)**

Calls the default editor.

### **reset**

Clears the query buffer.

### **quit or exit**

Exits `isql`.

The 5701 ("changed database") server message is no longer displayed after login or issuing a `use database` command.

Error message format differs from earlier versions of `isql`. If you have scripts that perform routines based on the values of these messages you may need to rewrite them.

When you include the `-x` parameter, the password-enabled connection proceeds according to server capabilities:

- If the server can handle both extended password and password encryption, extended password encryption negotiations are used.
- If the server can handle password encryption only, password encryption negotiations are used.
- If the server cannot handle password encryption or extended password encryption, the first connection attempt fails and the client attempts to reconnect using a plain text password.

To use `isql` interactively, give the command `isql` (and any of the optional parameters) at your operating system prompt. The `isql` program accepts SQL commands and sends them to Adaptive Server. The results are formatted and printed on standard output. Exit `isql` with `quit` or `exit`.

Terminate a command by typing a line beginning with the default command terminator `go` or another command terminator, if the `-c` parameter is used. You can follow the command terminator with an integer to specify how many times to run the command. For example, to execute this command 100 times, type:

```
select x = 1
go 100
```

The results display once at the end of execution.

If you enter an option more than once on the command line, `isql` uses the last value. For example, if you enter the following command, "send", the second value for `-c`, overrides ".", the first value:

```
isql -c"." -csend
```

This enables you to override any aliases you set up.

To call an editor on the current query buffer, enter its name as the first word on a line. Define your preferred callable editor by specifying it with the `EDITOR` environment variable. If `EDITOR` is not defined, the default is `vi` on Linux and `edit` on Windows.

For example, if your `EDITOR` environment variable is set to "emacs," then you must invoke it from within `isql` with "emacs" as the first word on the line.

To clear the existing query buffer, type `reset` on a line by itself. `isql` discards any pending input. You can also press `Ctrl-c` anywhere on a line to cancel the current query and return to the `isql` prompt.

Read in an operating system file containing a query for execution by `isql` as follows:

```
isql -U alma -P password < input_file
```

The file must include a command terminator. The results appear on your terminal. Read in an operating system file containing a query and direct the results to another file as follows

```
isql -U alma -P password < input_file > output_file
```

`isql` displays only six digits of float or real data after the decimal point, rounding off the remainder.

You can include comments in a Transact-SQL statement submitted to Adaptive Server by `isql`. Open a comment with `/*`. Close it with `*/`, as shown in the following example:

```
select au_lname, au_fname
/*retrieve authors' last and first names*/
from authors, titles, titleauthor
where authors.au_id = titleauthor.au_id
and titles.title_id = titleauthor.title_id
/*this is a three-way join that links authors
**to the books they have written.**
```

If you want to comment out a `go` command, it should not be at the beginning of a line. For example, use the following to comment out the `go` command:

```
/*
**go
*/
```

Do not use the following:

```
/* go */
```

`isql` defines the order of the date format as month, date, and year (mm dd yyyy hh:mmAM (or PM)) regardless of the locale environment. To change this default order, use the `convert` function.

In an `isql` session, the default prompt label is either the default wildcard `?:` or the value of `<wildcard>`. You can customize the prompt label by providing a one-word character string, with a maximum length of 80 characters after a wildcard. If you specify a prompt label that is more than one word, the characters after the first word are ignored.

### Note

In an `isql` session, `isql` recognizes `?:` or the value of `<wildcard>`, as wildcards only when these characters are placed at the beginning of an `isql` line.

## Examples

- Puts you in a text file where you can edit the query. When you write and save the file, you are returned to `isql`. The query appears; type "go" on a line by itself to execute it:

```
isql -Ujoe -Pabracadabra
1> select *
2> from authors
3> where city = "Oakland"
4> vi
```

- `reset` clears the query buffer. `quit` returns you to the operating system:

```
isql -Ualma
Password:
1> select *
2> from authors
3> where city = "Oakland"
4> reset
1> quit
```

- Specifies that you are running `isql` from a Macintosh against a server that is using the `roman8` character set:

```
isql -a mac -J roman8
```

- Creates column separators using the `"#"` character in the output in the `pubs2` database for store ID 7896:

```
isql -Usa -P -s#
1> use pubs2
2> go
1> select * from sales where stor_id = "7896"
#stor_id#ord_num          #date          #
#-----#-----#-----#
#7896   #124152          #      Aug 14 1986 12:00AM#
#7896   #234518          #      Feb 14 1991 12:00AM#
(2 rows affected)
```

- Changes password without displaying the password entered. This example uses "old" and "new" as prompt labels:

```
$ isql -Uguest -Pguest -Smyase --conceal
sp_password
:? old
'
:?:? new
-----
old
new
Confirm new
Password correctly set.
(Return status 0)
```

- Activates a role for the current user. This example uses a custom wildcard and the prompt labels "role" and "password:"

```
$ isql -UmyAccount --conceal '*'Password:
set role
* role
with passwd
** password
on
go
role
password
Confirm password
```

# 28 start\_iq Database Server Startup Utility

The database startup utility `start_iq` starts an SAP IQ network database server.

The UNIX versions of SAP IQ provide the script `start_iq`, and the Windows version provides `start_iq.exe`. These scripts verify that your environment is set correctly and start the server with all required switches preset to recommended defaults (along with any switches you add).

The `start_iq` utility also includes some parameters and calculates others. For switches that are specific to your operating system, see the *SAP IQ Installation and Update Guide*.

You set default parameters in `IQDIR16/scripts/default.cfg`. On Linux and UNIX, if you set the environment variable `IQ_NO_DEFAULTS` to `YES`, then the default parameters aren't included.

## ⚠ Caution

For security, SAP recommends you start the server using TLS or HTTPS communication server options.

### In this section:

#### [Starting the start\\_iq Utility \[page 101\]](#)

Start the `start_iq` database startup utility from the command line on Windows and UNIX.

#### [Listing All start\\_iq Switches \[page 101\]](#)

You can display a list of all available switches for the `<server-options>`, `<database-file>`, and `<database-options>` parameters.

#### [start\\_iq Server Options \[page 101\]](#)

Available switches for the `start_iq <server-options>` parameters.

#### [start\\_iq Database File Parameters \[page 208\]](#)

This topic lists the parameters of the database server/database file. You specify the database file after the server options in the command syntax.

#### [start\\_iq Database Options \[page 209\]](#)

Available switches for the `start_iq <database-options>` parameters.

## Related Information

[SAP IQ Installation and Update Guide for Linux](#)

[SAP IQ Installation and Update Guide for Windows](#)

[SAP IQ Installation and Update Guide for UNIX \(HP-UX, IBM-AIX, and Solaris\) \[Deprecated\]](#)

## 28.1 Starting the start\_iq Utility

Start the `start_iq` database startup utility from the command line on Windows and UNIX.

### Procedure

1. Issue a command in this format:

```
start_iq [ <server-options> ] [ <database-file>
[ <database-options> ], ... ]
```

2. Specify the `-n` switch in `<server-options>` to prevent you from unintentionally starting the wrong server.

## 28.2 Listing All start\_iq Switches

You can display a list of all available switches for the `<server-options>`, `<database-file>`, and `<database-options>` parameters.

### Procedure

Issue the following command:

```
start_iq -?
```

## 28.3 start\_iq Server Options

Available switches for the `start_iq <server-options>` parameters.

### In this section:

#### [@data Database Server Option \[page 107\]](#)

Reads in options from the specified environment variable or configuration file.

#### [@envvar Database Server Option \[page 109\]](#)

Reads in command-line switches from the supplied environment variable.

#### [@filename Database Server Option \[page 109\]](#)

Reads in switches from configuration file.

- al [Database Server Option \[page 110\]](#)  
Extends LOGIN\_MODE for LDAPUA only to a select number of users using Standard authentication.
- c [Database Server Option \[page 111\]](#)  
Sets initial memory reserved for caching catalog store pages and other server information.
- ca [Database Server Option \[page 111\]](#)  
Enforces a static cache size.
- cc [Database Server Option \[page 113\]](#)  
Collects information about database pages to be used for cache warming the next time the database is started.
- ch [Database Server Option \[page 114\]](#)  
Sets catalog store cache size upper limit in bytes.
- cl [Database Server Option \[page 115\]](#)  
Sets a minimum cache size as a lower limit to dynamic cache resizing.
- cp [Database Server Option \[page 116\]](#)  
Specifies a set of directories or JAR files in which to search for classes.
- cr [Database Server Option \[page 117\]](#)  
Reloads (warms) the cache with database pages using information collected the last time the database was run.
- cs [Database Server Option \[page 118\]](#)  
Displays statistics related to dynamic cache sizing in the database server messages window.
- ct [Database Server Option \[page 119\]](#)  
Sets an upper limit on the amount of memory consumed by table definition objects.
- cv [Database Server Option \[page 120\]](#)  
Controls the appearance of messages about cache warming in the database server messages window.
- dt [Database Server Option \[page 121\]](#)  
Specifies the directory where temporary files are stored.
- ec [Database Server Option \[page 123\]](#)  
Uses transport layer security (TLS) or simple obfuscation to encode communication protocol packets (such as DBLIB and ODBC) transmitted to and from all clients.
- ep [Database Server Option \[page 127\]](#)  
Prompts the user for the encryption key upon starting a strongly encrypted database.
- es [Database Server Option \[page 128\]](#)  
Allows unencrypted connections over shared memory.
- fips [Database Server Option \[page 129\]](#)  
Sets the database server to use FIPS-certified (FIPS PUB 140-2) encryption for strong database, SQL encryption and hashing functions, and communication encryption.
- ga [Database Server Option \[page 130\]](#)  
Unloads the database after the last non-HTTP client connection disconnects.
- gb [Database Server Option \[page 131\]](#)  
Sets the server process priority class.
- gc [Database Server Option \[page 132\]](#)  
Sets the maximum interval between checkpoints.

- gd Database Server Option [page 132]  
Sets the privileges required to start or stop a database on a running database server.
- ge Database Server Option [page 134]  
Sets the stack size for external functions.
- gf Database Server Option [page 135]  
Disables firing of triggers by the server.
- gk Database Server Option [page 135]  
Sets the privileges required to stop the database server.
- gl Database Server Option [page 136]  
Sets the permission required to load data using `LOAD TABLE`.
- gm Database Server Option [page 137]  
Limits the number of concurrent connections to the database server.
- gn Database Server Option [page 138]  
Sets the number of execution threads that are used for the catalog store and connectivity while running with multiple users.
- gp Database Server Option [page 139]  
Sets the maximum allowed database page size.
- gr Database Server Option [page 139]  
Sets the maximum length of time, in minutes, that the database server takes to recover from system failure.
- gss Database Server Option [page 140]  
Sets, in part, the stack size for server execution threads that execute requests for server connections.
- gt Database Server Option [page 140]  
Sets the maximum number of physical processors that can be used (up to the licensed maximum). This option is only useful on multiprocessor systems.
- gta Database Server Option [page 141]  
Sets which logical processors the database server can use.
- gtc Database Server Option [page 143]  
Controls the maximum processor concurrency that the database server allows.
- gu Database Server Option [page 144]  
Sets the privilege required for executing database file administration statements such as for creating or dropping databases.
- iqdirectio Database Server Option [page 146]  
Enables and disables direct I/O in your simplex server.
- iqfilereadahead Database Server Option [page 146]  
Enables or disables prefetching at the file system level.
- iqfreq Database Server Option [page 147]  
Opens database in forced recovery mode.
- iqgovern Database Server Option [page 147]  
Sets the number of concurrent queries allowed by the server.
- iqhrt Database Server Option [page 148]  
Enables high-resolution timing for the `iqmsg` IQ message log file.

- [-iqlm Database Server Option \[page 149\]](#)  
Specifies the maximum amount of memory in MB that SAP IQ can dynamically request from the operating system for temporary use.
- [-iqlrt Database Server Option \[page 149\]](#)  
Controls the number of threads that SAP IQ uses during point-in-time recovery operations.
- [-iqmc Database Server Option \[page 150\]](#)  
Specifies the main IQ store buffer cache size, in MB.
- [-iqmpx\\_failover Database Server Option \[page 151\]](#)  
Initiates multiplex coordinator failover to establish the designated failover secondary node as the new coordinator.
- [-iqmpx\\_ov Database Server Option \[page 151\]](#)  
Performs multiplex configuration override for the current node.
- [-iqmpx\\_reclaimwriterfreelist Database Server Option \[page 152\]](#)  
This option applies only while restarting a coordinator node. The coordinator forcefully reclaims the free list of the writer node identified by the server name.
- [-iqmpx\\_sn Database Server Option \[page 152\]](#)  
Initiates the multiplex. Runs the current multiplex node in single-node mode.
- [-iqmsgnum Database Server Option \[page 152\]](#)  
Specifies the number of archives of the old message log maintained by the server.
- [-iqmsgsz Database Server Option \[page 153\]](#)  
Limits the maximum size of the message log.
- [-iqmt Database Server Option \[page 153\]](#)  
Specifies the number of SAP IQ threads to create.
- [-iqnotemp Database Server Option \[page 154\]](#)  
Creates a temporary file in place of the defined temporary dbspace.
- [-iqnumbercpus Database Server Option \[page 155\]](#)  
This option provides an SAP IQ query optimizer hint specifying the degree of parallelism used by various query execution nodes within a query block.
- [-iqpartition Database Server Option \[page 156\]](#)  
Sets the number of IQ main and temp buffer cache partitions.
- [-iqlvmem Database Server Option \[page 156\]](#)  
Specifies the amount of memory, in megabytes, available to the RLV store.
- [-iqro Database Server Option \[page 157\]](#)  
Runs the current node read-only.
- [-iqstart Database Server Option \[page 157\]](#)  
Provides startup diagnostics for dbspaces.
- [-iqtc Database Server Option \[page 158\]](#)  
Specifies IQ temporary store cache size, in MB.
- [-iqtss Database Server Option \[page 159\]](#)  
Specifies the stack size, in KB, for server execution threads running either in the background or as part of a thread team assisting the main server connection thread.
- [-iqwmem Database Server Option \[page 159\]](#)

Creates a pool of “wired” memory on HP and Sun UNIX systems.

[-k Database Server Option \[page 160\]](#)

Controls the collection of Windows Performance Monitor statistics and statement performance summary statistics.

[-kl Database Server Option \[page 161\]](#)

Specifies the file name of the Kerberos GSS-API library (or shared object on UNIX and Linux) and enables Kerberos authenticated connections to the database server.

[-kr Database Server Option \(Deprecated\) \[page 161\]](#)

Specifies the realm of the Kerberos server principal and enables Kerberos authenticated connections to the database server.

[-krb Database Server Option \[page 162\]](#)

Enables Kerberos-authenticated connections to the database server.

[-ks Database Server Option \[page 163\]](#)

Disables the creation of shared memory that the Windows Performance Monitor uses to collect counter values from the database server.

[-ksc Database Server Option \[page 164\]](#)

Specifies the maximum number of connections that the Windows Performance Monitor can monitor.

[-ksd Database Server Option \[page 164\]](#)

Specifies the maximum number of databases that the Windows Performance Monitor can monitor.

[-lt Database Server Option \[page 165\]](#)

Sets the number of lock tables for each database.

[-m Database Server Option \[page 166\]](#)

Truncates the transaction log when a checkpoint is done.

[-mp Database Server Option \[page 167\]](#)

Controls the number of heap memory pools used by the server catalog cache.

[-n Database Server Option \[page 168\]](#)

Sets the name of the database server.

[-o Database Server Option \[page 169\]](#)

Prints all database server messages to the database server message log file.

[-oe Database Server Option \[page 170\]](#)

Specifies a file name to log startup errors, fatal errors, and assertions.

[-on Database Server Option \[page 170\]](#)

Specifies a maximum size for the database server message log, after which the file is renamed with the extension `.old` and a new file is started.

[-os Database Server Option \[page 171\]](#)

Specifies a maximum size for the database server message log file, at which point the file is renamed.

[-ot Database Server Option \[page 172\]](#)

Truncates the database server message log file and appends output messages to it.

[-p Database Server Option \[page 173\]](#)

Sets the maximum size of communication packets.

[-pc Database Server Option \[page 173\]](#)

Compresses all connections except for same-computer connections.

- [-phl Database Server Option \[page 174\]](#)  
Sets history tracking for specified database server properties.
- [-phs Database Server Option \[page 175\]](#)  
Sets the maximum amount of memory to use for tracking property history, in allotted time or bytes.
- [-pt Database Server Option \[page 176\]](#)  
Increases or decreases the size limit at which TCP/IP packets are compressed.
- [-qi Database Server Option \[page 177\]](#)  
Controls whether database server system tray icon and database server messages window appear.
- [-qp Database Server Option \[page 177\]](#)  
Specifies that messages about performance do not appear in the database server messages window.
- [-qs Database Server Option \[page 178\]](#)  
Suppresses startup error windows.
- [-qw Database Server Option \[page 178\]](#)  
Specifies that the database server messages window does not appear.
- [-s Database Server Option \[page 179\]](#)  
Sets the user ID for Syslog messages.
- [-sb Database Server Option \[page 179\]](#)  
Specifies how the database server reacts to broadcasts.
- [-sf Database Server Option \[page 180\]](#)  
Controls whether users have access to features for databases running on the current database server.
- [-sk Database Server Option \[page 186\]](#)  
Creates the SYSTEM secured feature key and sets the authorization code for it. This permits access to features that are secured for the database server.
- [-su Database Server Option \[page 188\]](#)  
Sets the user ID and password for the utility database (utility\_db), or disables connections to the utility database.
- [-tdsl Database Server Option \[page 189\]](#)  
Sets the TDS login mode.
- [-ti Database Server Option \[page 190\]](#)  
Disconnects inactive connections.
- [-tl Database Server Option \[page 191\]](#)  
Sets the period at which to send liveness packets.
- [-tmf Database Server Option \[page 192\]](#)  
Forces transaction manager recovery for distributed transactions.
- [-tq Database Server Option \[page 192\]](#)  
Shuts down the server at a specified time.
- [-u Database Server Option \[page 193\]](#)  
Opens files using the operating system disk cache.
- [-ud Database Server Option \[page 193\]](#)  
(UNIX servers) Causes the process to run as a daemon in the root directory. SAP recommends that you do not use this switch in IQ servers.
- [-uf Database Server Option \[page 194\]](#)

Specifies the action that the database server takes when a fatal error or assertion failure occurs on a database server.

[-ufd Database Server Option \[page 194\]](#)

Specifies the action that the database server takes when a fatal error or assertion failure occurs on a database.

[-ut Database Server Option \[page 196\]](#)

(UNIX servers) Causes the server to touch catalog store temporary files at intervals specified by `<min>`.

[-v Database Server Option \[page 196\]](#)

Displays the database server version in a message box (Windows) or in a version string (UNIX/Linux).

[-x Database Server Option \[page 196\]](#)

Specifies server-side network communications protocols.

[-xd Database Server Option \[page 198\]](#)

Prevents the database server from becoming the default database server.

[-xs Database Server Option \[page 198\]](#)

Specifies server-side Web services communications protocols.

[-z Database Server Option \[page 201\]](#)

Displays diagnostic communication messages, and other messages, for troubleshooting purposes.

[-ze Database Server Option \[page 201\]](#)

Displays database server environment variables in the database server messages window.

[-zl Database Server Option \[page 202\]](#)

Turns on capturing of the most recently prepared SQL statement for each connection to databases on the server.

[-zn Database Server Option \[page 203\]](#)

Specifies the number of request log file copies to retain.

[-zo Database Server Option \[page 204\]](#)

Redirects request logging information to a file separate from the regular log file.

[-zoc Database Server Option \[page 205\]](#)

Redirects web service client information to a file.

[-zr Database Server Option \[page 205\]](#)

Enables request logging of operations.

[-zs Database Server Option \[page 207\]](#)

Limits the size of the request log.

## 28.3.1 @data Database Server Option

Reads in options from the specified environment variable or configuration file.

### Syntax

```
start_iq @<data> ...
```

## Applies to

All operating systems.

This option is also supported for all database utilities except the Language Selection utility (dblang) and the Start Server in Background utility (dbspawn).

## Remarks

Use this option to read in command-line options from the specified environment variable or configuration file. If both exist with the same name that is specified, the environment variable is used.

Configuration files can contain line breaks, and can contain any set of options.

To protect the information in a configuration file (for example, because it contains passwords), use the File Hiding (dbfhide) utility to encrypt the contents of configuration files.

The @data parameter can occur at any point in the command, and parameters contained in the file are inserted at that point. Multiple files can be specified, and the file specifier can be used with command line options.

## Example

The following example starts a database server named **myserver** that loads the sample databases. The database server does not automatically adjust to the static cache size, which is 40% of the total physical memory:

```
-c 4096
-n myserver
"c:\mydatabase.db"
```

If this configuration file is saved as `c:\config.txt`, it can be used in a command as follows:

```
start_iq @c:\config.txt
```

The following configuration file contains comments:

```
#This is the server name:
-n MyServer
#These are the protocols:
-x tcpip
#This is the database file
my.db
```

The following statement sets an environment variable that holds options for a database server that starts with a cache size of 4 MB and loads a database called mydatabase.db.

```
SET envvar=-c 4096 "c:\mydatabase.db" ;
```

The following command starts the database server using an environment variable named **envvar**.

```
start_iq @envvar
```

## Related Information

[dbfhide Database Administration Utility \[page 27\]](#)

### 28.3.2 @envvar Database Server Option

Reads in command-line switches from the supplied environment variable.

#### ↳ Syntax

```
start_iq @envvar
```

#### Remarks

The environment variable may contain any set of command-line switches. For example, the first of the following pair of Windows statements sets an environment variable holding a set of command-line switches and loads the sample database. The second statement starts the database server:

```
set envvar= -gp 4096 -gm 15  
/IQ/IQ-16_1/demo/start_iq -n myserver @envvar iqdemo.db
```

#### ⓘ Note

If you have both a file and an environment variable with the value of your @ command line switch, the result is unpredictable. Use only one of these methods to set a given @ command line switch.

### 28.3.3 @filename Database Server Option

Reads in switches from configuration file.

#### ↳ Syntax

```
start_iq @<filename>
```

## Remarks

The file may contain line breaks, and may contain any set of command-line switches. For example, the following Windows command file holds a set of command-line switches for a server named `iqdemo` that allows 10 connections, and sets the maximum catalog page size to 4096 bytes, and starts the `iqdemo` database:

```
# iqdemo.cfg
# -----
# Default startup parameters for the IQ demo database
# -----
-n iqdemo
-x tcpip{port=2638}
# The following parameters are also found in the configuration file
# %IQDIR%\scripts\default.cfg. Any parameters not specified below
# and not in the start up parameter list, will be added by start_iq
# using default.cfg as a guide.
-c 48m
-gc 20
-gd all
-gl all
-gm 10
-gp 4096
-iqmc 32
-iqtc 24
```

If this configuration file is saved as `c:\config.txt`, the file can be used in a command line as follows:

```
start_iq @c:\config.txt
```

## Related Information

[Configuration File List](#)

### 28.3.4 -al Database Server Option

Extends LOGIN\_MODE for LDAPUA only to a select number of users using Standard authentication.

#### ↔ Syntax

```
start_iq -al "<user1>;<user2>;<user3>" <server_name>.cfg <database-name>.db
```

## Remarks

- Up to five user IDs can be specified, separated by semi-colons, and enclosed in double quotation marks.
- When run at the server level, the `-al` switch remains in effect until the next time the server is restarted.

## 28.3.5 -c Database Server Option

Sets initial memory reserved for caching catalog store pages and other server information.

### ↳ Syntax

```
start_iq -c <cache-size> [ k | m | g | p ]
```

### Remarks

The database server uses extra memory for caching database pages if the memory is set aside in the cache. Any cache size less than 10000 is assumed to be KB (1K = 1024 bytes). Any cache size 10000 or greater is assumed to be in bytes. You can also specify the cache size nK, nM, or nP (1M = 1024 KB), where P is a percentage of the physical system memory.

In the `default.cfg` file and `start_iq`, the default value of `-c` is as follows:

- Windows – 32 MB
- UNIX platforms – 48 MB

Use these defaults or set `-c` to a higher value.

You can use % as an alternative to P, but as most non-UNIX operating systems use % as an environment variable escape character, you must escape the % character. For example, to use 20 percent of the physical system memory, specify:

```
start_iq -c 20%% ...
```

### Related Information

[-ct Database Server Option \[page 119\]](#)

[-ch Database Server Option \[page 114\]](#)

[-ca Database Server Option \[page 111\]](#)

[-cl Database Server Option \[page 115\]](#)

## 28.3.6 -ca Database Server Option

Enforces a static cache size.

### ↳ Syntax

```
start_iq -ca 0 ...
```

## Applies to

All operating systems.

## Remarks

You can disable automatic cache size tuning by specifying `-ca 0` option. If you do not include the `-ca 0` option, the database server automatically increases the cache size. If you specify this option, the cache size is still adjusted if the database server would otherwise run into an error indicating that the dynamic memory is exhausted.

This server option must only be used in the form `-ca 0`.

## Example

The following example starts a database server named `myserver` that has a static cache that is 40% of the total physical memory and loads the sample database, and the database server does not automatically adjust the cache size:

The following example starts a database server named `myserver` that has a static cache that is 40% of the total physical memory and loads the sample database, and the database server does not automatically adjust the cache size:

```
start_iq @"%
  ALLUSERSPROFILE%\SybaseIQ\demo\iqdemo.cfg" -c 40P
  -ca 0 -n myservers "%
  ALLUSERSPROFILE%\SybaseIQ\demo\iqdemo.db"
```

The following example is for UNIX/Linux:

```
start_iq
  @$IQDIR16/demo/iqdemo.cfg -c 40P -ca 0
  $IQDIR16/demo/iqdemo.db
```

## Related Information

- [-c Database Server Option \[page 111\]](#)
- [-cc Database Server Option \[page 113\]](#)
- [-ch Database Server Option \[page 114\]](#)
- [-cl Database Server Option \[page 115\]](#)
- [-cr Database Server Option \[page 117\]](#)
- [-cs Database Server Option \[page 118\]](#)
- [-cv Database Server Option \[page 120\]](#)

## 28.3.7 -cc Database Server Option

Collects information about database pages to be used for cache warming the next time the database is started.

### ↔ Syntax

```
start_iq -cc{ + | - } ...
```

### Default

By default, page collection is turned on.

### Applies to

All operating systems.

### Allowed Values

- cc+ Turns on page collection.
- cc- Turns off page collection.

### Remarks

When collection is turned on, the database server keeps track of each database page that is requested. Collection stops when the maximum number of pages has been collected, the database is shut down, or the collection rate falls below the minimum value. You cannot configure the maximum number of pages collected or specify the value for the collection rate (the value is based on cache size and database size). Once collection stops, information about the requested pages is recorded in the database so those pages can be used to warm the cache the next time the database is started with the -cr option. Collection of referenced pages is turned on by default.

### Related Information

- [-c Database Server Option \[page 111\]](#)
- [-ca Database Server Option \[page 111\]](#)
- [-ch Database Server Option \[page 114\]](#)

- [-cl Database Server Option \[page 115\]](#)
- [-cr Database Server Option \[page 117\]](#)
- [-cs Database Server Option \[page 118\]](#)
- [-cv Database Server Option \[page 120\]](#)

## 28.3.8 -ch Database Server Option

Sets catalog store cache size upper limit in bytes.

### ↔ Syntax

```
start_iq -ch <size> [ k | m | g | p ]
```

### Remarks

By default, the upper limit is approximately the lower of 256 MB and 90 percent of the physical memory of the machine.

You specify the cache-size using the K, M, and P characters as in the `-c` database server option. For the meaning and usage of the cache size argument and the K, M, and P characters, see *-c Database Server Option*.

In some cases the standard catalog cache size may be too small, for example, to accommodate certain queries that need a lot of parsing. In these cases, you may find it helpful to set `-c1` and `-ch`. For example, on 32-bit platforms, try these settings:

```
-c1 128M  
-ch 512M
```

### ⚠ Caution

To control catalog store cache size explicitly, you must do either of the following, but not both, in your configuration file (`.cfg`) or on the UNIX command line for server startup:

- Set the `-c` parameter.
- Set specific upper and lower limits for the catalog store cache size using the `-ch` and `-c1` parameters.

Specifying different combinations of the parameters above can produce unexpected results.

### Related Information

- [-c Database Server Option \[page 111\]](#)
- [-cl Database Server Option \[page 115\]](#)

## 28.3.9 -cl Database Server Option

Sets a minimum cache size as a lower limit to dynamic cache resizing.

### Syntax

```
start_iq -cl <size>[ k | m | g | p ] ...
```

### Default

2 MB on Windows

8 MB on UNIX and Linux

### Applies to

All operating systems.

### Remarks

This option sets a lower limit to the cache. If `-c` is specified, and `-cl` is not specified, then the minimum cache size is set to the initial cache size (the `-c` setting). If neither `-c` nor `-cl` is set, the minimum cache is set to a low, constant value, so that the cache can shrink if necessary. On Windows platforms, this value is 2 MB.

The `<size>` is the amount of memory, in bytes. Use `k`, `m`, or `g` to specify units of kilobytes, megabytes, or gigabytes, respectively.

The unit `p` is a percentage either of the physical system memory, or of the maximum supported cache size, whichever is lower. The maximum supported cache size depends on the operating system. For example:

- 2.5 GB for Windows 32-bit Advanced Server, Enterprise Server, and Datacenter Server
- 3.5 GB for the 32-bit database server running on Windows x64 Edition
- 1.5 GB on all other 32-bit systems
- On 64-bit database servers, the cache size can be considered unlimited

If you use `p`, the argument is a percentage. You can use `%` as an alternative to `p`, but on Windows, which uses `%` as an environment variable escape character, you must escape the `%` character.

For example, to set the minimum cache size to 50 percent of the physical system memory, run the following command:

```
start_iq -cl 50%% ...
```

### Note

If you attempt to set your initial or minimum cache sizes to a value that is less than one eighth of the maximum cache size, the initial and minimum cache sizes are automatically increased relative to the supported cache size.

## Example

The following example starts a database server named silver that has a minimum cache size of 5 MB and loads the database file `example.db`:

```
start_iq -cl 5m -n silver "c:\example.db"
```

## Related Information

- [-c Database Server Option \[page 111\]](#)
- [-ca Database Server Option \[page 111\]](#)
- [-cc Database Server Option \[page 113\]](#)
- [-ch Database Server Option \[page 114\]](#)
- [-cr Database Server Option \[page 117\]](#)
- [-cs Database Server Option \[page 118\]](#)
- [-cv Database Server Option \[page 120\]](#)

## 28.3.10 -cp Database Server Option

Specifies a set of directories or JAR files in which to search for classes.

### Syntax

```
start_iq -cp <location>[ ;<location> ... ] ...
```

On UNIX and Linux, quotation marks are required for the `-cp` value when more than one `<location>` is supplied.

On Windows, quotation marks are required for the `-cp` value when space characters appear in `<location>`.

## Applies to

All operating systems.

## Remarks

All classes and JAR files that are being used with Java in the database must be installed in the database. When you store the classes and JAR files within the database, the database can be easily moved to a different computer or operating system. Another benefit of installing classes and JAR files into the database is that the database server's class loader can fetch the classes and resources from the database, allowing each connection that is using Java in the database to have its own instance of these classes and its own copy of static variables within these classes.

However, when a class or JAR file must be loaded by the system class loader, it can be specified with the `java_class_path` database option or the `-cp` database server option. Both options add classes and JAR files to the classpath that the database server builds for launching the Java VM. The database server does not use the `CLASSPATH` as defined by the environment variable of that name. The `java_class_path` database option is useful when the server is running multiple databases and each database has a different set of JARs and directories that need to be loaded by the system class loader. The `-cp` database server option is useful when all databases on the server require the same classes or JAR files.

The JAR files required to support the database server as an OData server are always also included in the server's class path.

## Related Information

[Java in the Database](#)

### 28.3.11 -cr Database Server Option

Reloads (warms) the cache with database pages using information collected the last time the database was run.

#### Syntax

```
start_iq -cr{ + | - } ...
```

## Default

By default, cache warming is turned on.

## Applies to

All operating systems.

## Allowed Values

- cr+ Turns on cache warming.
- cr Turns on cache warming.
- cr- Turns off cache warming.

## Remarks

You can instruct the database server to warm the cache using pages that were referenced the last time the database was started (page collection is turned on using the -cc option). Cache warming is turned on by default. When a database is started, the server checks the database to see if it contains a collection of pages requested the last time the database was started. If the database contains this information, the previously referenced pages are then loaded into the cache.

Warming the cache with pages that were referenced the last time the database was started can improve performance when the same query or similar queries are executed against a database each time it is started.

## Related Information

- [-cc Database Server Option \[page 113\]](#)
- [-cl Database Server Option \[page 115\]](#)
- [-cs Database Server Option \[page 118\]](#)
- [-cv Database Server Option \[page 120\]](#)

## 28.3.12 -cs Database Server Option

Displays statistics related to dynamic cache sizing in the database server messages window.

### ↳ Syntax

```
start_iq -CS ...
```

## Applies to

All operating systems.

## Remarks

For troubleshooting purposes, this option displays statistics in the database server messages window that database server is using to determine how to tune size of the cache.

## Related Information

- [-c Database Server Option \[page 111\]](#)
- [-ca Database Server Option \[page 111\]](#)
- [-cc Database Server Option \[page 113\]](#)
- [-ch Database Server Option \[page 114\]](#)
- [-cl Database Server Option \[page 115\]](#)
- [-cr Database Server Option \[page 117\]](#)
- [-cv Database Server Option \[page 120\]](#)

## 28.3.13 -ct Database Server Option

Sets an upper limit on the amount of memory consumed by table definition objects.

### ⌘ Syntax

```
start_iq -ct <size>
```

The minimum value of `<size>` is 512 KB.

## Remarks

The SAP IQ server maintains an in-memory data structure that represents the state of tables. When a table is first created, its definition structure gets added to this data structure, where it remains, until the table is dropped.

The server retains all table definition objects in memory even if the tables are no longer referenced, resulting in the server consuming large amounts of memory when a database contains thousands of tables.

Use `-ct <size>` to set a limit on the amount of memory that table definition objects can consume.

When the total memory consumed by table definition objects exceeds the limit set by `-ct <size>`, the server starts evicting from its memory those tables that are no longer referenced.

### 📌 Note

The server does not evict tables as long as they are referenced by active connections, even when the table definition objects consume more than `<size>`.

Alternatively, you can set the limit at runtime by using the MaxTableDefCacheSize server option:

```
CALL sa_server_option( MaxTableDefCacheSize , <size> );
```

A DBA can use the MaxTableDefCacheSize server property to retrieve the current set limit:

```
SELECT PROPERTY ( 'MaxTableDefCacheSize' );
```

A MaxTableDefCacheSize value of 0 indicates that no limit is imposed.

To find out how much memory is currently being consumed by table definition objects, use the CurrentTableDefCacheSize database property with the DB\_PROPERTY system function:

```
SELECT DB_PROPERTY ( 'CurrentTableDefCacheSize' );
```

## Related Information

[-c Database Server Option \[page 111\]](#)

[List of Database Server Properties](#)

[sa\\_server\\_option System Procedure](#)

## 28.3.14 -cv Database Server Option

Controls the appearance of messages about cache warming in the database server messages window.

### ⌘ Syntax

```
start_iq -cv{ + | - } ...
```

## Default

Cache warming messages are suppressed.

## Applies to

All operating systems.

## Allowed Values

- cv+ Cache warming messages are displayed.
- cv Cache warming messages are displayed.
- cv- Cache warming messages are suppressed.

## Remarks

When -cv+ is specified, a message appears in the database server messages window when any of the following cache warming activities occur:

- collection of requested pages starts or stops (controlled by the -cc server option)
- page reloading starts or stops (controlled by the -cr server option)

## Example

The following command starts the database `mydatabase.db` with database page collection and page loading turned on, and logs messages about these activities to the database server messages window:

```
start_iq -cc+ -cr+ -cv+ mydatabase.db
```

## Related Information

- [-c Database Server Option \[page 111\]](#)
- [-ca Database Server Option \[page 111\]](#)
- [-cc Database Server Option \[page 113\]](#)
- [-ch Database Server Option \[page 114\]](#)
- [-cl Database Server Option \[page 115\]](#)
- [-cr Database Server Option \[page 117\]](#)
- [-cs Database Server Option \[page 118\]](#)

## 28.3.15 -dt Database Server Option

Specifies the directory where temporary files are stored.

### ⌘ Syntax

```
start_iq -dt <temp-file-dir> ...
```

## Applies to

All operating systems, except shared memory connections on UNIX and Linux.

## Remarks

The database server creates two types of temporary files:

### Database server-related temporary files (created on all platforms)

You can use the `-dt` option to specify a directory for database server-related temporary files. If you do not specify this option when starting the database server, the database server examines the following environment variables, in the order shown, to determine the directory in which to place the temporary file.

#### Windows/UNIX/Linux

---

1. IQTMP16
  2. TMP
  3. TMPDIR
  4. TEMP
- 

If none of the environment variables are defined, the database server places its temporary file in the current directory on Windows, and in the `/tmp` directory on UNIX and Linux.

### Communications-related temporary files (created only on UNIX and Linux for both the client and the database server)

Temporary files for communications on UNIX and Linux are not placed in the directory specified by `-dt`. Instead, the database server examines the following environment variables, in the order shown, to determine the directory in which to place the temporary file:

#### UNIX/Linux

---

1. IQTMP16
  2. TMP
  3. TMPDIR
  4. TEMP
- 

If none of the environment variables are defined, the database server places its temporary file in the `/tmp` directory on UNIX and Linux.

On UNIX and Linux, both the client and the database server must set IQTMP16 to the same value when connecting via shared memory.

## Example

To locate the database server-related temporary files, use the DB\_PROPERTY system function with the TempFileName property:

```
SELECT DB_PROPERTY ( 'TempFileName' );
```

## Related Information

[sa\\_disk\\_free\\_space System Procedure](#)

## 28.3.16 -ec Database Server Option

Uses transport layer security (TLS) or simple obfuscation to encode communication protocol packets (such as DBLIB and ODBC) transmitted to and from all clients.

### Syntax

```
start_iq -ec <encoding-option> [,...]
```

Specify at least one of the supported options in a comma-separated list.

```
<encoding-option> ::=
{ NONE
 | SIMPLE
 | TLS (
   [ FIPS={ ON | OFF } ; ]
   IDENTITY=<server-identity-filename>;
   IDENTITY_PASSWORD=<password>
   [ ; MIN_TLS_VERSION=<ver> ]
   [ ; other_network_protocol_option=<value> ]
   SET_CIPHERS=<ciphers_string>
 )
```

On UNIX and Linux, quotation marks are required when specifying the TLS options for the -ec option:

```
start_iq -ec "TLS(<encoding-option> [;...])"
```

## Allowed Values

### NONE

The database server accepts connections that aren't encrypted. These connections can be remote connections, or they can be local connections over shared memory.

### SIMPLE

The database server accepts connections that are encoded using simple obfuscation. The database server also accepts connections that aren't encoded and upgrades them to simple obfuscation.

When both the SIMPLE and NONE parameters are specified, the database server accepts connections that aren't encoded, but doesn't upgrade them.

## TLS

The database server accepts connections that are encrypted using the Transport Layer Security RSA algorithm. The TLS parameter accepts the following arguments:

### FIPS

For FIPS-certified TLS, specify FIPS=ON. FIPS-certified TLS uses a separate library, but is compatible with uncertified TLS.

### IDENTITY=server-identity-filename

The path and file name of the server identity certificate. You must generate your certificates using the RSA algorithm.

### IDENTITY\_PASSWORD=password

The password for the private key contained in the identity file. This password was defined when the server identity certificate was created.

### MIN\_TLS\_VERSION=ver

The minimum downgrade TLS version. The supported values for `<ver>` are [1.0](#), [1.1](#), [1.2](#), and [1.3](#)). TLS 1.2 and 1.3 are selected by default with the server and client negotiating the highest protocol they both support. The `min_tls_version` network protocol option can be used to change allowed protocols. The period is optional, so you can also specify [10](#), [11](#), [12](#), or [13](#).

### other\_network\_protocol\_option=value

```
other_network_protocol_option :=
{
  TRUSTED_CERTIFICATE
  | CERTIFICATE_COMPANY
  | CERTIFICATE_NAME
  | CERTIFICATE_UNIT
  | SET_CIPHERS
  | SKIP_CERTIFICATE_NAME_CHECK }

```

### SET\_CIPHERS

Specifies either a security level of the allowed algorithms, or a list of specified cipher suites.

Define the `<ciphers_string>` in this format:

```
set_ciphers=<ccl_cipher_selector_group> ( :: <ccl_curve_selector_group> ) *
ccl_cipher_selector_group := [!] <cipher_spec> ( : [!] <cipher_spec> ) *
cipher_spec := <cipher_alias> | <cipher_name>

```

`<ccl_cipher_selector_group>` is the cipher suite group such as [PFS](#) (perfect forward secrecy cipher suites). `<ccl_curve_selector_group>` is the elliptic curve cipher suite group such as [EC\\_HIGH](#) (high strength elliptic curve cipher suites).

`<cipher_alias>` is one of the aliases defined in [TLS Cipher Suite Aliases](#).

`<cipher_name>` is one of the names defined in [TLS Support](#).

The `<ciphers_string>` uses these conventions:

- `:` separates configuration items (within a group)

- :: separates different groups
- ! removes items from the result list

## Default

The default is NONE, SIMPLE.

When both the SIMPLE and NONE parameters are specified, the database server accepts connections that are encoded using simple obfuscation as well as connections that aren't encoded.

## Applies To

Applies to all operating systems.

## Remarks

To fully secure the server, clients and server should mutually trust each others' certificate chains and ensure that only secure communication is accepted by the server. Clients should perform checks on the server's certificate, via the host name check or the name/company/unit checks. Self-signed server and client certificates are not recommended. See Using Client-side Certificates for TLS for more details on how to setup mutual trust.

Use this option to secure communication packets between client applications and the database server. By default, communication packets aren't encrypted, which poses a potential security risk. If you're concerned about the security of network packets, then use the -ec option.

The -ec option instructs the database server to accept only connections that are encrypted using one of the specified types. Specify at least one of the supported parameters in a comma-separated list.

The -ec NONE option instructs the database server to accept both remote and local shared memory connections that aren't encrypted.

The -es SIMPLE option uses simple obfuscation that doesn't provide server verification, strong encryption, or other features of transport layer security.

If you specify -ec TLS or -ec SIMPLE and you want to accept shared memory connections (which aren't encoded/encrypted), then you must also specify the -es database server option.

All external environments connect back to the database server over shared memory. If the database server is configured to accept only encrypted connections, then the external environment will fail to connect. In this case, make sure to include the -es option with the other database server start options to enable shared memory connections.

The client's and the database server's encryption settings must match or the connection fails except in the following cases:

- If `-ec SIMPLE` is specified on the database server, but `-ec NONE` isn't, then connections that don't specify the Encryption connection parameter and connections that specify `Encryption=NONE` can connect and are automatically upgraded to use simple obfuscation.
- If the database server specifies uncertified TLS encryption and the client specifies FIPS-certified TLS encryption, or vice versa, then the connection succeeds. In these cases, the Encryption connection property returns the value specified by the database server.

### ⚠ Caution

To fully secure your server, you must also specify TDS communication behavior (see [-x Database Server Option \[page 196\]](#) and [TDS Communication Parameter](#)). Choose `TDS=NO` to disallow TDS connections or `TDS=ENCRYPTED` to accept only TLS encrypted TDS connections. TDS communication (such as Java applications using `jConnect`, or the SAP ASE `isql` tool) does not support SIMPLE encryption.

Encryption affects performance only marginally.

The `dbrsa17.dll` (`libdbrsa17_r.so` on UNIX and Linux) file contains the TLS code used for encryption and decryption. The `dbfips17.dll` (`libdbfips17_r.so` on Linux) file contains the FIPS-certified version of the TLS algorithm. When you connect to the database server, if the appropriate file can't be found, or if an error occurs, then a message appears in the database server messages window. The database server doesn't start if the specified types of encryption can't be initiated.

## Example

Start a secure server:

```
-x tcpip(port=1234;TDS=ENCRYPTED) -ec  
"TLS(identity=myserver.id;identity_password=XXXXXX) "
```

The following example starts a database server that uses TLS encryption and the identity certificate `rsaserver.id`.

On Windows:

```
start_iq -ec TLS(IDENTITY=rsaserver.id;IDENTITY_PASSWORD=test) -x  
tcpip(TDS=ENCRYPTED) mydemo.db
```

On UNIX and Linux:

```
start_iq -ec "TLS(IDENTITY=rsaserver.id;IDENTITY_PASSWORD=test)" -x  
"tcpip(TDS=ENCRYPTED)" mydemo.db
```

The following Windows example starts a database server using FIPS-certified TLS encryption algorithms and the identity certificate `rsaserver.id`. If `rsaserver.id` is expired, then it still accepted by the database server. Note that TDS connections are disabled in this example.

```
start_iq -ec TLS(FIPS=ON;IDENTITY=rsaserver.id;IDENTITY_PASSWORD=test); -x tcpip  
(TDS=NONE) mydemo.db
```

The following example connects to the mydemo server and displays the text `rsa_tls_fips` at the command prompt.

```
dbisql -c
"UID=DBA;PWD=passwd;ENC=TLS(TRUSTED_CERTIFICATE=rsaroot.crt;SKIP_CERTIFICATE_NAME
_CHECK=ON);Server=mydemo;LINKS=TCPIP" select connection_property('encryption')
```

### Note

The sample certificates referenced here are installed in the `samples\certificates` folder.

This Windows example starts a database server that uses `TLS_ECDHE_RSA_WITH_ARIA_128_GCM_SHA256` and `TLS_ECDHE_RSA_WITH_ARIA_256_GCM_SHA384` encryption, and uses the identity certificate `rsaserver.id`:

```
start_iq -ec
TLS(SET_CIPHERS=TLS_ECDHE_RSA_WITH_ARIA_128_GCM_SHA256:TLS_ECDHE_RSA_WITH_ARIA_25
6_GCM_SHA384;IDENTITY=rsaserver.id;IDENTITY_PASSWORD=test) -x tcpip mydemo.db
```

This Windows example starts a database server, selects all ciphers that have perfect forward secrecy (PFS) strength, rejects all mSHA1 and mSHA256 ciphers, and uses the identity certificate `rsaserver.id`:

```
start_iq -ec TLS(SET_CIPHERS=PFS:!mSHA1:!
mSHA256;IDENTITY=rsaserver.id;IDENTITY_PASSWORD=test) -x tcpip mydemo.db
```

## Related Information

[Encryption \(ENC\) Connection Parameter](#)

[Network Communications Parameters](#)

[TDS Communication Parameter](#)

[-x Database Server Option \[page 196\]](#)

[Digital Certificates](#)

[Encryption \(ENC\) Connection Parameter](#)

[-ep Database Server Option \[page 127\]](#)

[-es Database Server Option \[page 128\]](#)

[-fips Database Server Option \[page 129\]](#)

[Using Client-side Certificates for TLS](#)

## 28.3.17 -ep Database Server Option

Prompts the user for the encryption key upon starting a strongly encrypted database.

### Syntax

```
start_iq -ep ...
```

## Applies to

All operating systems.

## Remarks

The `-ep` option instructs the database server to display a window where the user enters the encryption key for databases started on the command line that require an encryption key. This server option provides an extra measure of security by never allowing the encryption key to be seen in clear text.

When used with the database server, the user is prompted for the encryption key when the following are all true:

- The `-ep` option is specified
- An encryption key is required to start a database
- The database server isn't a Windows service or a daemon (UNIX and Linux)

To secure communication packets between client applications and the database server, use the `-ec` server option and transport layer security.

## Example

The user is prompted for the encryption key when the `myencrypted.db` database is started:

```
start_iq -ep -x tcpip myencrypted.db
```

## 28.3.18 -es Database Server Option

Allows unencrypted connections over shared memory.

### ↔ Syntax

```
start_iq -ec <encryption-options> -es ...
```

## Applies to

All operating systems.

## Remarks

This option is only effective when specified with the `-ec` option. The `-es` option instructs the database server to accept local, unencrypted, command-sequence connections over shared memory. In contrast the `-ec NONE` option instructs the database server to accept both remote and local, command-sequence connections that are not encrypted.

Connections over TCP/IP must use an encryption type specified by the `-ec` option. The `-es` option is useful in situations where you want remote clients to use encrypted connections, but for performance reasons you also want to access the database from the local computer with an unencrypted connection.

## Example

- The following example specifies that connections with simple obfuscation and unencrypted connections over shared memory are allowed.

```
start_iq -ec SIMPLE -es -x tcpip c:\mydemo.db
```

## 28.3.19 -fips Database Server Option

Sets the database server to use FIPS-certified (FIPS PUB 140-2) encryption for strong database, SQL encryption and hashing functions, and communication encryption.

### ↔ Syntax

```
start_iq -fips ...
```

## Applies to

Windows and Linux.

## Remarks

This option applies to strong database encryption, client/server transport layer security, and web services transport layer security.

For strong database encryption, the `-fips` option causes new databases to use the FIPS-certified equivalent of AES and AES256 if they are specified in the `ALGORITHM` clause of the `CREATE DATABASE` statement.

When the database server is started with `-fips`, you can run databases encrypted with AES, AES256, AES\_FIPS, or AES256\_FIPS encryption, but not databases encoded with simple obfuscation. Unencrypted databases can also be started on the server when `-fips` is specified.

The SAP IQ FIPS-certified AES/RSA encryption components must be installed on any computer used to run a database encrypted with AES\_FIPS or AES256\_FIPS.

For transport layer security, the `-fips` option causes the server to use the FIPS-certified encryption algorithm, even if RSA is specified.

For transport layer security for web services, the `-fips` option causes the server to use FIPS-certified HTTPS, even if HTTPS is specified instead of HTTPS\_FIPS.

When you specify `-fips`, the ENCRYPT and HASH functions use the FIPS-certified encryption algorithm. For example, if you specify SHA1, then SHA1\_FIPS is used instead.

When you specify `-fips`, password hashing uses the SHA256\_FIPS algorithm instead of the SHA256 algorithm.

Password hashing refers to the way passwords are stored in the database. It does not refer to how passwords are exchanged/encrypted between client and server. In this case, the AES256/AES256\_FIPS encryption algorithm is used.

When you specify `-fips` and then you start an email session using the `xp_startsmtp` system procedure, then the database server always uses PLAIN authentication to authenticate the IDs and passwords with the SMTP server. The `-fips` database server option does not support CRAM\_MD5 authentication.

## Related Information

[Simple Obfuscation Versus Strong Encryption](#)  
[-ec Database Server Option \[page 123\]](#)

## 28.3.20 -ga Database Server Option

Unloads the database after the last non-HTTP client connection disconnects.

### Syntax

```
start_iq -ga ...
```

## Applies to

All operating systems.

## Remarks

Specifying this option on the network server causes each database to be unloaded after the last non-HTTP client connection disconnects. In addition to unloading each database after the last non-HTTP connection disconnects, the database server shuts down when the last database is stopped.

If the only connection to a database is an HTTP connection, and the database is configured to stop automatically, when the HTTP connection disconnects, the database is not unloaded. As well, if you specify the `-ga` option, and the database has an HTTP connection and a command sequence or TDS connection, when the last command sequence or TDS connection disconnects, the database stops automatically, and any HTTP connections are dropped.

## 28.3.21 -gb Database Server Option

Sets the server process priority class.

### ⌘ Syntax

(Windows only)

```
start_iq -gb { IDLE | NORMAL | HIGH | MAXIMUM } ...
```

(UNIX/Linux only)

```
start_iq -gb <level> ...
```

## Allowed Values

### Windows

On Windows, `NORMAL` and `HIGH` are the commonly used settings. The value `IDLE` is provided for completeness. The value `MAXIMUM` may interfere with the running of your computer.

### UNIX/Linux

On UNIX and Linux, `<level>` is an integer from -20 to 19. The default value on UNIX and Linux is the same as the nice value of the parent process. Lower `<level>` values represent a more favorable scheduling priority. All restrictions placed on setting a nice value apply to the `-gb` option. For example, on most UNIX and Linux platforms, only the root user can lower the priority level of a process (for example, changing it from 0 to -1).

## Applies to

All operating systems.

## 28.3.22 -gc Database Server Option

Sets the maximum interval between checkpoints.

### ↳ Syntax

```
start_iq -gc <minutes> ...
```

### Default

60 minutes

### Allowed Values

#### minutes

The default value is the setting of the `checkpoint_time` database option, which defaults to 60 minutes. If a value of 0 is entered, the default value of 60 minutes is used.

### Applies to

All operating systems.

### Remarks

Use this option to set the maximum length of time, in minutes, that the database server runs without doing a checkpoint on each database.

Checkpoints generally occur more frequently than the specified time.

## 28.3.23 -gd Database Server Option

Sets the privileges required to start or stop a database on a running database server.

### ↳ Syntax

```
start_iq -gd { DBA | ALL | NONE } ...
```

## Allowed Values

### DBA

Only users with the SERVER OPERATOR system privilege can start or stop databases.

### ALL

All users can start or stop databases. Not recommended for network servers that can be accessed by remote clients.

### NONE

Starting and stopping databases isn't allowed except when the database server itself is started and stopped.

## Default

The default setting is *DBA* for the network database server. Both uppercase and lowercase syntax are allowed.

## Applies to

All operating systems.

## Remarks

This option specifies the level of privilege required for a user to cause a new database file to be loaded by the database server, or to stop a database on a running database server.

When the option is set to DBA, the client application must use an existing connection to another database running on the same server to start and stop databases. You cannot start a database that is not already running by using the DatabaseFile (DBF) connection parameter.

You can obtain the setting of the -gd option using the StartDBPermission server property:

```
SELECT PROPERTY ( 'StartDBPermission' );
```

The privileges for stopping a database server are specified by the -gk option.

## Example

The following steps illustrate how to use the -gd option for the network database server.

1. Start the network database server:

```
start_iq -su passwd -gd DBA -n my_server
```

2. Connect to the utility database from Interactive SQL:

```
dbisql -c "UID=DBA;PWD=passwd;DBN=utility_db"
```

3. Start a database:

```
START DATABASE 'iqdemo.db';
```

4. Connect to the database:

```
CONNECT USING 'DBN=demo;UID=DBA;PWD=sql';
```

## 28.3.24 -ge Database Server Option

Sets the stack size for external functions.

### ↳ Syntax

```
start_iq -ge <integer> ...
```

### Default

32 KB

### Applies to

Windows.

### Remarks

Sets the stack size for threads running external functions, in bytes.

## 28.3.25 -gf Database Server Option

Disables firing of triggers by the server.

### ↳ Syntax

```
start_iq -gf ...
```

### Applies to

All operating systems.

### Remarks

The -gf server option instructs the server to disable the firing of triggers, including referential integrity triggers (such as cascading updates and deletes).

## 28.3.26 -gk Database Server Option

Sets the privileges required to stop the database server.

### ↳ Syntax

```
start_iq -gk { DBA | ALL | NONE } ...
```

### Allowed Values

#### DBA

Only users with the SERVER OPERATOR system privilege can stop the database server.

This value is the default.

#### ALL

No privileges are required to shut down the database server.

#### NONE

The database server cannot be stopped.

## Applies to

All operating systems.

## Remarks

The `-gd` database server option applies to the `dbstop` utility as well as to the following statements:

- ALTER DATABASE `<dbname>` FORCE START statement.
- STOP DATABASE statement

## 28.3.27 -gl Database Server Option

Sets the permission required to load data using `LOAD TABLE`.

### ⌘ Syntax

```
start_iq -gl <level>
```

## Remarks

The `LOAD TABLE` statement reads files from the database server machine. To control access to the file system using these statements, `-gl <level>` allows you to control the level of database privileges that are required to use these statements, as follows:

Level	Description
DBA	Only users with the <code>LOAD ANY TABLE</code> , <code>ALTER ANY TABLE</code> or <code>ALTER ANY OBJECT</code> system privilege can load data.
ALL	Only users with one of the following: <ul style="list-style-type: none"><li>• You are the owner of the table</li><li>• ALTER object-level privilege on the table</li><li>• LOAD object-level privilege on the table</li><li>• ALTER ANY TABLE system privilege</li><li>• LOAD ANY TABLE system privilege</li><li>• ALTER ANY OBJECT system privilege</li></ul>
NONE	Data cannot be loaded.

The option is case insensitive.

The default setting is `<ALL>` for servers started with `start_iq` and `<DBA>` for all other servers. For consistency with earlier versions, use `<ALL>` on all systems. `<ALL>` is used in the `iqdemo.cfg` and `default.cfg` configuration files.

## 28.3.28 -gm Database Server Option

Limits the number of concurrent connections to the database server.

### Syntax

```
start_iq -gm <integer> ...
```

### Default

The default value and the connection limit is 10 connections.

### Applies to

All operating systems.

### Remarks

Defines the connection limit for the database server.

The database server allows one extra connection above the connection limit to allow a user with the DROP CONNECTION or SERVER OPERATOR system privilege to connect to the database server and drop other connections.

The database server queues HTTP/HTTPS connection attempts when the database server reaches its HTTP/HTTPS connection limit. While there are connections in the HTTP/HTTPS queue, there is no opportunity for a standard connection to replace an HTTP/HTTPS connection. A user wanting to make a standard connection could wait indefinitely unless some of the database connections are reserved for standard connections.

Reserve some of the database's connections for standard connections by specifying the `reserved_connections` database option.

### Maximum Values

Platform	Maximum Value of -gm
UNIX and Linux	1001

Platform	Maximum Value of -gm
Windows	1001

You can also limit the maximum number of HTTP/HTTPS connections that can be made to a database server by specifying the MaxConnections protocol option.

## 28.3.29 -gn Database Server Option

Sets the number of execution threads that are used for the catalog store and connectivity while running with multiple users.

### ⌘ Syntax

```
start_iq -gn <integer>
```

### Remarks

This parameter applies to all operating systems and servers. Each connection uses a thread for each request, and when the request is completed, the thread is returned to the pool for use by other connections. As no connection can have more than one request in progress at one time, no connection uses more than one thread at a time.

An exception to this rule is if a Java application uses threads. Each thread in the Java application is a database server execution thread.

On Windows, specify this parameter in `start_iq`. To calculate its value, use:

```
<gn_value> >= <gm_value> * 1.5
```

Set the `-gn` value to at least 1.5 times the value of `-gm`. Specify a minimum of 25. The total number of threads cannot exceed a platform-specific maximum; see `-iqmt <num>` for details.

### Related Information

[-iqmt Database Server Option \[page 153\]](#)

## 28.3.30 -gp Database Server Option

Sets the maximum allowed database page size.

### ↔ Syntax

```
start_iq -gp { 2048 | 4096 | 8192 | 16384 | 32768 } ...
```

### Default

4096 (if a database server is started with no databases loaded).

### Applies to

All operating systems.

### Remarks

Database files with a page size larger than the page size of the server cannot be loaded. This option explicitly sets the page size of the server, in bytes.

By default, the server page size is the same as the largest page size of the databases on the command line.

The 2048 page size is deprecated.

On all platforms, if you do not use this option and start a server with no databases loaded, the default value is 4096.

## 28.3.31 -gr Database Server Option

Sets the maximum length of time, in minutes, that the database server takes to recover from system failure.

### ↔ Syntax

```
start_iq -gr <num>
```

## 28.3.32 -gss Database Server Option

Sets, in part, the stack size for server execution threads that execute requests for server connections.

### ↳ Syntax

```
start_iq -gss {<integer> | <integerK> | <integerM>}
```

### Remarks

SAP IQ calculates the stack size of these server threads using the formula:  $(-gss + -igtss)$ . See `-igtss`.

On UNIX, the default value for `-gss` is 1 MB. The maximum stack size is 8 MB.

On Windows, the default value for `-gss` is 4 MB. The maximum stack size is 256 MB.

### Related Information

[-igtss Database Server Option \[page 159\]](#)

## 28.3.33 -gt Database Server Option

Sets the maximum number of physical processors that can be used (up to the licensed maximum). This option is only useful on multiprocessor systems.

### ↳ Syntax

```
start_iq -gt <num-processors>...
```

### Allowed Values

#### **num-processors**

This integer can be a value between 1 and the minimum of:

- the number of potential physical processors on the computer, and
- the maximum number of CPUs that the server is licensed for if CPU-licensing is in effect.

If the `-gt` value specified lies outside this range, the minimum or maximum integer value is imposed.

## Remarks

With per-seat licensing, the network database server uses all CPUs available on the computer. With CPU-based licensing, the network database server uses only the number of processors you are licensed for. The number of CPUs that the network database server can use may also be restricted by your SAP IQ edition.

Physical processors are sometimes referred to as packages or dies, and are the CPUs of the computer. Additional logical processors exist when the physical processors support hyperthreading or are themselves configured as multiprocessors (often referred to as multi-core processors). The operating system schedules threads on logical processors.

When you specify a value for the `-gt` option, the database server adjusts its affinity mask (if supported on that hardware platform) to restrict the database server to run on only that number of physical processors. If the database server is licensed for `<n>` processors, the server, by default, runs on all logical processors (hyperthreads and cores) of `<n>` physical processors. This behavior can be further restricted with the `-gtc` option.

## 28.3.34 -gta Database Server Option

Sets which logical processors the database server can use.

### ↳ Syntax

```
start_iq -gta <logical-processors-to-use>,...
```

On UNIX and Linux, quotation marks are required when more than one parameter is supplied.

## Allowed Values

### logical-processors-to-use

Accepts a comma-delimited list of processor numbers and/or ranges. A range is indicated with a hyphen. If the lower endpoint of a range is omitted, then it is assumed to be zero. If the upper endpoint of a range is omitted, then it is assumed to be the highest CPU known to the operating system. To specify all processors type `-` or `0-`.

## Applies to

Microsoft Windows and Linux.

## Remarks

Setting this option is equivalent to executing the ProcessorAffinity option of the [sa\\_server\\_option System Procedure](#).

The -gta option allows you to specify which logical processors are used by the database server.

The database server might not use all of the specified logical processors in the following cases:

- If one or more of the specified logical processors does not exist, or is offline.
- If the license does not allow it.

If the set of logical processors specified by the -gta option exceeds the license limits, the server uses the lowest-numbered logical processors specified by the -gta option up to, but not exceeding, the license limits. The database server does not choose logical processors in the order listed by the -gta option. The server does not attempt to maximize concurrency within the license limit and the specified -gta option.

If the set of logical processors specified by the -gta option does not match any online processors, the server acts as if the -gta option was not specified and uses up to the licensed number of logical processors, starting at processor 0.

### ⚠ Restriction

The -gta option cannot be used with the [-gt Database Server Option \[page 140\]](#) or the [-gtc Database Server Option \[page 143\]](#).

While the database server is running, you can change which logical processors it uses with the ProcessorAffinity option of the [sa\\_server\\_option System Procedure](#).

## Example

The following example requires the database server to use some or all of the logical processors 3, 5, 6, 7, and/or 9:

```
start_iq -gta 3,5-7,9
```

## Related Information

[-iqnumbercpus Database Server Option \[page 155\]](#)

[-gt Database Server Option \[page 140\]](#)

[-gtc Database Server Option \[page 143\]](#)

## 28.3.35 -gtc Database Server Option

Controls the maximum processor concurrency that the database server allows.

### ⌘ Syntax

```
start_iq -gtc <logical-processors-to-use> ...
```

### Applies to

Windows, Linux, and Solaris operating systems running on Intel-compatible x86 and x64 platforms.

### Remarks

When you start the database server, the number of physical and logical processors detected by the database server appears in the database server messages window.

Physical processors are sometimes referred to as packages or dies, and are the CPUs of the computer. Additional logical processors exist when the physical processors support hyperthreading or are themselves configured as multiprocessors (often referred to as multi-core processors). The operating system schedules threads on logical processors.

The -gtc option allows you to specify the number of logical processors that can be used by the database server. Its effect is to limit the number of database server threads that are created at server startup. This limits the number of active database server tasks that can execute concurrently at any one time. By default, the number of threads created is 1 + the number of logical processors on all licensed physical processors.

By default, the database server allows concurrent use of all logical processors (cores or hyperthreads) on each licensed physical processor.

For example, when using a one-CPU license on a two-CPU system where each CPU contains four cores with two threads per core, the network database server permits eight threads to run concurrently on one CPU and zero threads on the other. If the -gtc option is specified, and the number of logical processors to be used is less than the total available for the number of physical processors that are licensed, then the database server allocates logical processors based on round-robin assignment. Specifying 1 for the -gtc option implicitly disables intra-query parallelism (parallel processing of individual queries). Intra-query parallelism can also be explicitly limited or disabled using the max\_query\_tasks option.

### Example

Consider the following examples for a Windows-based SMP computer. In each case, assume a 4-processor system with two cores on each physical processor for a total of eight logical processors. The physical processors are identified with letters and the logical processors (cores in this case) are identified with numbers. This 4-processor system therefore has processing units A0, A1, B0, B1, C0, C1, D0, and D1.

Scenario	Network database server settings
A single CPU license or -gt 1 specified	<ul style="list-style-type: none"> <li>-gt 1</li> <li>-gtc 2</li> </ul> <p>Threads can execute on A0 and A1.</p>
No licensing restrictions on the CPU with -gtc 5 specified	<ul style="list-style-type: none"> <li>-gt 4</li> <li>-gtc 5</li> </ul> <p>Threads can execute on A0, A1, B0, C0, and D0.</p>
A database server with a three CPU license and -gtc 5 specified	<ul style="list-style-type: none"> <li>-gt 3</li> <li>-gtc 5</li> </ul> <p>Threads can execute on A0, A1, B0, B1, and C0.</p>
No licensing restrictions on the CPU with -gtc 1 specified	<ul style="list-style-type: none"> <li>-gt 4</li> <li>-gtc 1</li> </ul> <p>Threads can execute only on A0.</p>

## Related Information

[-gt Database Server Option \[page 140\]](#)

### 28.3.36 -gu Database Server Option

Sets the privilege required for executing database file administration statements such as for creating or dropping databases.

#### ↔ Syntax

```
start_iq -gu { DBA | ALL | NONE | utility_db } ...
```

## Allowed Values

-gu option	Effect	Applies to
<i>DBA</i>	Only users with the SERVER OPERATOR system privilege can execute file administration statements	Any database including utility database
<i>ALL</i>	<i>This option is deprecated.</i> Anyone can execute file administration statements.	Any database including utility database

<b>-gu option</b>	<b>Effect</b>	<b>Applies to</b>
<i>NONE</i>	Executing file administration statements is not allowed.	Any database including utility database
<i>utility_db</i>	Only the users who can connect to the utility database can execute file administration statements	Only the utility database

## Default

DBA

## Remarks

Restricts the users who can execute the following database file administration statements:

- ALTER DATABASE dbfile ALTER TRANSACTION LOG
- CREATE DATABASE statement
- DROP DATABASE statement
- RESTORE DATABASE statement.

When *utility\_db* is specified, these statements can only be run from the utility database. When *DBA* is specified, these statements can only be run by a user with the `SERVER OPERATOR` system privilege. When none is specified, no user can execute these statements.

## Example

To prevent the use of the file administration statements, start the database server using the `none` privilege level of the `-gu` option. The following command starts a database server and names it `TestSrv`. It loads the `mytestdb.db` database, but prevents anyone from using that server to create or delete a database, or execute any other file administration statement regardless of their resource creation rights, or whether they can load and connect to the utility database.

```
start_iq -n TestSrv -gu none c:\mytestdb.db
```

To permit only the users knowing the utility database password to execute file administration statements, start the server by running the following command.

```
start_iq -n TestSrv -su passwd -gu utility_db
```

The following command starts Interactive SQL as a client application, connects to the server named `TestSrv`, loads the utility database, and connects the user.

```
dbisql -c "UID=DBA;PWD=passwd;DBN=utility_db;Host=host1;Server=TestSrv"
```

Having executed the above command successfully, the user connects to the utility database, and can execute file administration statements.

## 28.3.37 -iqdirectio Database Server Option

Enables and disables direct I/O in your simplex server.

### ↔ Syntax

```
start_iq -iqdirectio < ON | OFF >
```

### Default

ON

### Applies to

All platforms.

Simplex only. Ignored in multiplex.

### Remarks

When enabled, SAP IQ uses direct I/O in your simplex server wherever possible.

SAP strongly recommends you do not set `-iqdirectio` to `OFF`. Keep direct I/O enabled in your simplex server to ensure optimal performance.

## 28.3.38 -iqfilereadahead Database Server Option

Enables or disables prefetching at the file system level.

### ↔ Syntax

```
start_iq -iqfilereadahead < ON | OFF >
```

## Default

OFF

## Applies to

All platforms.

Simplex only. Ignored in multiplex.

Buffered I/O (not direct I/O).

## Remarks

Prefetching speeds up data access by reading file contents from a prefetched cache, rather than from disk.

Requires buffered I/O. Buffered I/O only applies when you have disabled direct I/O (or if the underlying file system does not support direct I/O).

SAP strongly recommends you do not set `-iqdirectio` to `OFF`. Keep direct I/O enabled in your simplex server to ensure optimal performance.

## 28.3.39 -iqfrec Database Server Option

Opens database in forced recovery mode.

### ↳ Syntax

```
start_iq -iqfrec <dbname>
```

## 28.3.40 -iqgovern Database Server Option

Sets the number of concurrent queries allowed by the server.

### ↳ Syntax

```
start_iq -iqgovern <num>
```

## Default

The default value of `-iqgovern` is  $(2 \times \text{the number of CPUs}) + 10$ .

## Remarks

The number of concurrent queries is not the same as the number of connections. A single connection can have multiple open cursors. `-iqgovern` can help SAP IQ optimize paging of buffer data out to disk and avoid overcommitting memory. The default value of this switch is equal to 2 times the number of CPUs on your machine, plus 10. You may find that another value, such as 2 times the number of CPUs plus 4, provides better throughput, especially when large numbers of users are connected.

## 28.3.41 -iqhrt Database Server Option

Enables high-resolution timing for the `iqmsg` IQ message log file.

### ↔ Syntax

```
start_iq -iqhrt <ON>
```

## Default

OFF

## Remarks

This option enables high-resolution timing in the IQ message log file to the millisecond.

The default is OFF. There is no OFF parameter — to turn the option off, do not include it in the `start_iq` command line.

## 28.3.42 -iq1m Database Server Option

Specifies the maximum amount of memory in MB that SAP IQ can dynamically request from the operating system for temporary use.

### ⌘ Syntax

```
start_iq -iq1m <size>
```

### Default

2048MB

### Remarks

Some load operations may require more large memory than the 2 GB default provides. If memory requirements exceed the default, use the `-iq1m` startup option to increase the memory that SAP IQ can dynamically request from the OS. Set `-iq1m` as a switch as part of the command or configuration file that starts the server.

As a general rule, large memory requirements represent one third of the total available physical memory allocated to SAP IQ. To ensure adequate memory for the main and temporary IQ stores, set the `-iq1m`, `-iqtc`, and `-iqmc` startup parameters so that each parameter receives one third of all available physical memory allocated to SAP IQ.

In most cases, you should allocate 80 percent of total physical memory to SAP IQ to prevent SAP IQ processes from being swapped out. Adjust actual memory allocation to accommodate other processes running on the same system. For example, on a machine with 32 cores and 128 GB of total available physical memory, you would allocate 100 GB (approximately 80 percent of the 128 GB total) to SAP IQ processes. Following the general rule, you would set the `-iq1m`, `-iqtc`, and `-iqmc` parameters to 33 GB each.

### 📌 Note

Always specify the size value, without including the units of measurement. For example, specify `-iq1m 2048` rather than `-iq1m 2048MB`. If you specify the unit of measurement, `start_iq` ignores this switch.

## 28.3.43 -iq1rt Database Server Option

Controls the number of threads that SAP IQ uses during point-in-time recovery operations.

### ⌘ Syntax

```
start_iq -iq1rt <number>
```

## Default

50

## Remarks

The default is 50 threads per CPU. The total number of all server threads cannot exceed 4096.

## 28.3.44 -iqmc Database Server Option

Specifies the main IQ store buffer cache size, in MB.

### ⌘ Syntax

```
start_iq -iqmc <size>
```

## Default

64 MB

## Remarks

The switch overrides the default value of 64 MB, and applies to all databases started from the time the server is started until the server is shut down. In other words, if you start one database at server startup and another later, you need  $2 * -iqmc$  available for the main buffer cache.

Large memory requirements represent one third of all available physical memory. To ensure adequate memory for the main store, set the `-iqmc` startup parameter to one third of all available physical memory.

Always specify the size value, without including the units of measurement; for example, specify `-iqmc 32` rather than `-iqmc 32MB`. If you specify the unit of measurement, `start_iq` ignores this switch, unlike SAP SQL Anywhere, which requires a unit of measurement.

### ⓘ Note

Do not run multiple databases with an SAP IQ server.

## 28.3.45 -iqmpx\_failover Database Server Option

Initiates multiplex coordinator failover to establish the designated failover secondary node as the new coordinator.

### ❏ Syntax

```
iqmpx_failover 1
```

### Remarks

Starting the coordinator with this option has no effect. Users must be licensed for the Multiplex Grid Option to run secondary nodes.

## 28.3.46 -iqmpx\_ov Database Server Option

Performs multiplex configuration override for the current node.

### ❏ Syntax

```
-iqmpx_ov 1
```

### Remarks

Used to change node properties during startup in the event that a node's location or other property has changed. Initiates multiplex coordinator failover to establish the designated failover secondary node as the new coordinator. Starting the coordinator with this option has no effect. Users must be licensed for the Multiplex Grid Option to run secondary nodes.

### Related Information

[Server Startup Utility \(start\\_iq\)](#)

## 28.3.47 -iqmpx\_reclaimwriterfreelist Database Server Option

This option applies only while restarting a coordinator node. The coordinator forcefully reclaims the free list of the writer node identified by the server name.

### ⌘ Syntax

```
-iqmpx_reclaimwriterfreelist <server name>
```

### Remarks

This switch is only needed in the event that a writer fails and cannot be restarted. Users must be licensed for the Multiplex Grid Option to run secondary nodes.

## 28.3.48 -iqmpx\_sn Database Server Option

Initiates the multiplex. Runs the current multiplex node in single-node mode.

### ⌘ Syntax

```
-iqmpx_sn 1
```

### Remarks

This mode is used exclusively for fixing problems with the multiplex configuration and should be used with extreme caution. Requires all other nodes in the multiplex to be shut down. Recommended only for use on the coordinator node. Users must be licensed for the Multiplex Grid Option to run secondary nodes..

## 28.3.49 -iqmsgnum Database Server Option

Specifies the number of archives of the old message log maintained by the server.

### ⌘ Syntax

```
start_iq -iqmsgnum <num>
```

## Remarks

Allowed values are integers 0 – 64 (inclusive). Default value is 0, which means that messages are wrapped in the main message log file. Takes effect only if `-iqmsgnum` or the `IQMsgMaxSize` server property is a value other than zero. The `IQMsgNumFiles` server property corresponds to `-iqmsgnum` and takes precedence over the value of `-iqmsgnum`.

A `-iqmsgnum` value `<n>` greater than 0 means that the server maintains `<n>` message log archives. For example, this command specifies that the server maintain 3 archives of the message log:

```
start_iq -n iqdemo iqdemo.db ... <other options> ... -iqmsgsz 100 -iqmsgnum 3
```

## 28.3.50 -iqmsgsz Database Server Option

Limits the maximum size of the message log.

### ⌘ Syntax

```
start_iq -iqmsgsz <size>
```

## Remarks

`-iqmsgsz` is an integer 0-2047 (inclusive), in MB. The default value is 0, which specifies that there is no limit on message log size. The `IQMsgMaxSize` server property corresponds to the `-iqmsgsz` server switch and takes precedence over the value of `-iqmsgsz`.

A `-iqmsgsz` value `<n>` greater than 0 means that the message log can grow up to `<n>` megabytes in size. For example, the following command limits the size of the message log to 100 MB:

```
start_iq -n iqdemo iqdemo.db ... <other options> ... -iqmsgsz 100
```

## 28.3.51 -iqmt Database Server Option

Specifies the number of SAP IQ threads to create.

### ⌘ Syntax

```
start_iq -iqmt <num>
```

## Remarks

`<num>` is the number of threads. The default is 60 threads per CPU for the first 4 CPUs and 50 threads per CPU for the remainder, with 3 more for system use, plus threads needed for database connections and background tasks. For example, on a system with 12 CPUs and 10 connections:  $60 * 4 + 50 * (\text{<number\_of\_CPUs> - 4}) + \text{<number\_of\_connections>} + 6 = 656$ .

The minimum value of `<num>` is  $\text{<number\_of\_connections>} + 3$ .

The total number of server threads can't exceed 4096.

### → Tip

To reduce the risk of thread starvation, set `<num>` as close as possible to the maximum permissible value: 4096 for most 64-bit platforms. When calculating the maximum value, note that the **total** number of threads (`-iqmt` plus `-gn`) can't exceed the number allowed for your platform.

## Related Information

[-gn Database Server Option \[page 138\]](#)

[SAP Note 2936708](#)

## 28.3.52 -iqnotemp Database Server Option

Creates a temporary file in place of the defined temporary dbspace.

### ↗ Syntax

```
start_iq -iqnotemp <size>
```

## Remarks

`<size>` is file size, in MB. This parameter causes the server to ignore all SAP IQ temporary dbfile definitions when starting a database. You can use `-iqnotemp` to solve temporary dbfile problems by dropping damaged files and replacing them later.

If you start the utility database server with `-iqnotemp` when restoring, SAP IQ ignores all temporary file definitions in the backed up database during the restore operation, including intermediate starts. You can thus restore a database to a different temporary file topology without recreating and using old temporary file definitions.

The only temporary file operation allowed on a database started with `-iqnotemp` is:

```
ALTER DBSPACE IQ_SYSTEM_TEMP DROP FILE
```

## 28.3.53 -iqnumbercpus Database Server Option

This option provides an SAP IQ query optimizer hint specifying the degree of parallelism used by various query execution nodes within a query block.

### ↔ Syntax

```
start_iq -iqnumbercpus <num>
```

### Remarks

This option doesn't limit the number of CPUs or cores that are available to the SAP IQ server. The decision for how many CPUs to use is determined by the operating system, or by the `-gta` or `-gt` options. This option merely controls how SAP IQ optimizes a job, and a degree of parallelism for that job. For example, assume that your system has 20 CPUs. If you set `-iqnumbercpus` to 10, then SAP IQ optimizes for 10 – but could use all the way up to 20 CPUs. That is, a single query containing multiple query nodes may utilize a larger number of CPUs concurrently than what you set in `-iqnumbercpus`.

To specify a hard limit for the number of CPUs, use the [-gt Database Server Option \[page 140\]](#).

To control which CPUs SAP IQ can access, use the [-gta Database Server Option \[page 141\]](#).

The value of `-iqnumbercpus` defaults to the total number of CPUs, but the range of available values is 1–512.

Use `-iqnumbercpus` on these machines only:

- Machines with Intel CPUs and hyperthreading enabled, setting `-iqnumbercpus` to the number of CPU cores available
- Machines where an operating system utility has been used to restrict SAP IQ to a subset of the CPUs within the machine

Setting `-iqnumbercpus` higher than the number of available CPUs may affect performance.

### Related Information

[-gt Database Server Option \[page 140\]](#)

[-gta Database Server Option \[page 141\]](#)

[-gtc Database Server Option \[page 143\]](#)

## 28.3.54 -iqpartition Database Server Option

Sets the number of IQ main and temp buffer cache partitions.

### ↳ Syntax

```
-iqpartition <num>
```

### Remarks

Specifies the number of partitions in the IQ main and temp buffer caches. Must be a power of 2.

The default is 0, which means SAP IQ computes the number of partitions automatically according to this algorithm: `<number_of_cpus>`, rounded to the nearest power of 2, up to a maximum of 256.

By default, allowed values are: 0 (default), 1, 2, 4, 8, 16, 32, 64, 128, 256. You may be able to improve performance by adjusting the number of cache partitions. The `-iqpartition` switch sets this value for an SAP IQ server, and overrides the value set by the `CACHE_PARTITIONS` database option.

### Related Information

[CACHE\\_PARTITIONS Option](#)

## 28.3.55 -iqrvmem Database Server Option

Specifies the amount of memory, in megabytes, available to the RLV store.

### ↳ Syntax

```
start_iq -iqrvmem <size>
```

### Default

2048 (megabytes)

## Remarks

If you specify 0 or an invalid value, then the default (2048 MB) is used. If the value exceeds two-thirds of the system virtual memory limit, an error message appears, and the server shuts down.

## Usage

`-iqrlvmem` is used at server startup to tell the server how much memory to reserve for row-level versioning.

## 28.3.56 -iqro Database Server Option

Runs the current node read-only.

### ↔ Syntax

```
iqstart -iqro <1>
```

## Remarks

Used to verify simplex and multiplex migration and in legacy database unloads. Always specify the Boolean `<1>` (the only valid parameter) as an argument to this option.

## 28.3.57 -iqstart Database Server Option

Provides startup diagnostics for dbspaces.

### ↔ Syntax

```
start_iq -iqstart <N>
```

## Remarks

The input parameter `<N>` is a number value that represents an integer bit mask. You may combine values to provide more than one feature. Output generated before the IQ message file is generated goes to the console. The `-z` startup switch provides additional startup and connection information.

The allowed values are as follows:

Value	Description
N=1	Returns basic information about the file names from <code>SYSIQFILES</code> that are used when opening the dbspace. It then displays the fully qualified names used. You can use this option to create a record of the files in use by the database in the IQ message file.
N=2	Stops after the transaction log replay before executing <code>RecoveryComplete</code> allowing you to examine the database without opening it all the way. You can combine <code>&lt;N=2&gt;</code> with other options. In certain modes using <code>&lt;N=2&gt;</code> may rewrite the <code>commit_identity</code> , but does not otherwise modify the database in a permanent manner—the checkpoint that would commit the recovery actions is not allowed to complete. All recovery actions reexecute the next time the database is opened.
N=4	Returns full diagnostic information, including all rows of <code>SYSIQFILE</code> , the subset of file names selected if the database is a multiplex database, the fully resolved file names, each individual dbspace file header block, the <code>database_identity</code> , the <code>commit_identity</code> , each checkpoint log entry, and each transaction log entry.
N=8	Allows the file paths in <code>SYSIQFILE</code> to be overridden. Instead of the <code>SYSIQFILE</code> values, the file names <code>iqmsg.iqmsg</code> , <code>iqmain_1</code> , <code>iqmain_2</code> , ..., <code>iqtemp_1</code> , <code>iqtemp_2</code> , and so on. will be used. These may be links and must be in the same directory as the <code>.db</code> file. You may use a link to the actual <code>.db</code> file but if the server is given a link to a <code>.db</code> file that uses a transaction log relative to the database, the server looks for the transaction log relative to the link, rather than to the database. In this case, create a link for the transaction log also

## 28.3.58 -iqtc Database Server Option

Specifies IQ temporary store cache size, in MB.

### ⌘ Syntax

```
start_iq -iqtc <size>
```

### Default

64 MB

### Remarks

The switch overrides the default value of 64 MB, and applies to all databases started from the time the SAP IQ server is started until the server is shut down. In other words, if you start one database at server startup and another later, you need 2 \* `-iqtc` available for the temp cache. In general, avoid running multiple databases with an SAP IQ server.

Large memory requirements represent one third of all available physical memory. To ensure adequate memory for the IQ temporary store cache, set the `-iqtc` startup parameter to one third of all available physical memory.

Always specify the size value, without including the units of measurement; for example, specify `-iqtc 32` instead of `-iqtc 32MB`. If you specify the unit of measurement, `start_iq` ignores this switch, unlike SAP SQL Anywhere, which requires a unit of measurement.

Ensure that the IQ\_SYSTEM\_TEMP dbspace is at least as large as `-iqtc`.

## 28.3.59 -iqtss Database Server Option

Specifies the stack size, in KB, for server execution threads running either in the background or as part of a thread team assisting the main server connection thread.

### ⌘ Syntax

```
start_iq -iqtss <size>
```

### Remarks

The default is 512 KB on 64-bit platforms, and 200KB on 32-bit platforms.

### Related Information

[-gss Database Server Option \[page 140\]](#)

## 28.3.60 -iqwmem Database Server Option

Creates a pool of “wired” memory on HP and Sun UNIX systems.

### ⌘ Syntax

```
start_iq -iqwmem <size>
```

## Remarks

This memory is locked down so it cannot be paged by the operating system. Specify the memory size, in MB. Use this switch only if you have enough memory to dedicate for this purpose. Otherwise, you may cause serious performance degradation.

## 28.3.61 -k Database Server Option

Controls the collection of Windows Performance Monitor statistics and statement performance summary statistics.

### ↔ Syntax

```
start_iq -k ...
```

## Default

Windows Performance Monitor statistics and statement performance summaries are collected.

## Applies to

All operating systems.

## Remarks

If you specify -k when you start the database server, then the database server does not collect statistics for the Windows Performance Monitor. The -k option does not affect the collection of column statistics used by the query optimizer. You can also change the settings for the collection of Windows Performance Monitor statistics and statement performance data using the `sa_server_option` system procedure.

Additionally, this option disables statement performance summary collection. To disable statement performance summary collection on a single database, set the `CollectStmtPerfStats` option of the `sa_db_option` system procedure. Setting this option requires MONITOR privileges.

Statement performance data is collected only if both the database server and the database have data collection enabled.

This option should only be used in situations where the database server is running on a multi-processor computer where it can be shown by testing to improve performance. For most workloads, the benefit will be negligible, so use of this option is not recommended. When you disable the performance counters, this information is not available for analyzing performance problems.

## 28.3.62 -kl Database Server Option

Specifies the file name of the Kerberos GSS-API library (or shared object on UNIX and Linux) and enables Kerberos authenticated connections to the database server.

### ↔ Syntax

```
start_iq -kl <GSS-API-library-file> ...
```

### Applies to

All operating systems.

### Remarks

This option specifies the location and name of the Kerberos GSS-API. This option is only required if the Kerberos client uses a different file name for the Kerberos GSS-API library than the default, or if there are multiple GSS-API libraries installed on the computer running the database server. A Kerberos client must already be installed and configured, and SSPI cannot be used by the database server.

Specifying this option enables Kerberos authentication to the database server.

### Example

The following command starts a database server that uses the `libgssapi_krb5.so` shared object for Kerberos authentication.

```
start_iq -kl libgssapi_krb5.so -n my_server_princ /opt/myapp/kerberos.db
```

## 28.3.63 -kr Database Server Option (Deprecated)

Specifies the realm of the Kerberos server principal and enables Kerberos authenticated connections to the database server.

### ⓘ Note

The use of the `-kr` option is deprecated. Use the `-kp` option to specify the Kerberos server principal. If you specify `-kp`, the server principal must have been extracted to the Kerberos keytab file on the computer running the database server.

### ⌘ Syntax

```
start_iq -kr <server-realm> ...
```

## Applies to

All operating systems.

## Remarks

This option specifies the realm of the Kerberos server principal. Normally, the principal used by the database server for Kerberos authentication is `<server-name>@<default-realm>`, where `<default-realm>` is the default realm configured for the Kerberos client. Use this option if you want the server principal to use a different realm than the default realm, in which case the server principal used is `<server-name>@<server-realm>`.

Specifying this option enables Kerberos authentication to the database server.

The `-kr` option cannot be specified if the `-kp` option is specified.

## Example

The following command starts a database server that accepts Kerberos logins and uses the principal `my_server_princ@MYREALM` for authentication.

```
start_iq -kr MYREALM -n my_server_princ C:\kerberos.db
```

## 28.3.64 -krb Database Server Option

Enables Kerberos-authenticated connections to the database server.

### ⌘ Syntax

```
start_iq -krb ...
```

## Applies to

All operating systems.

## Remarks

This option enables Kerberos user authentication to the database server. You must specify one or more of the `-krb`, `-kl`, and `-kr` options for the database server to authenticate clients using Kerberos.

Before you can use Kerberos user authentication, a Kerberos client must already be installed and configured on both the client and database server computers. Additionally, the principal `<server-name>@<REALM>` must already exist in the Kerberos KDC, and the keytab for the principal `<server-name>@<REALM>` must already have been securely extracted to the keytab file on the database server computer. The database server will not start if the `-krb` option is specified, but this setup has not been performed.

### Note

The database server name cannot contain any of the following characters: `/`, `\`, or `@`, and database server names with multibyte characters cannot be used with Kerberos.

The `login_mode` database option must be set to allow Kerberos logins, and Kerberos client principals must be mapped to database user IDs using the `GRANT KERBEROS LOGIN` statement.

## Example

For a Kerberos principal for the database server named `my_server_princ@MYREALM`, the following command starts a database server named `my_server_princ`.

```
start_iq -krb -n my_server_princ C:\kerberos.db
```

## 28.3.65 -ks Database Server Option

Disables the creation of shared memory that the Windows Performance Monitor uses to collect counter values from the database server.

### Syntax

```
start_iq -ks 0 ...
```

## Applies to

Windows.

## Remarks

When you specify this option, the Windows Performance Monitor does not show any database server, database, or connection statistics for the current database server.

## 28.3.66 -ksc Database Server Option

Specifies the maximum number of connections that the Windows Performance Monitor can monitor.

### ↔ Syntax

```
start_iq -ksc <integer> ...
```

## Default

10

## Applies to

Windows.

## 28.3.67 -ksd Database Server Option

Specifies the maximum number of databases that the Windows Performance Monitor can monitor.

### ↔ Syntax

```
start_iq -ksd <integer> ...
```

## Default

2

## Applies to

Windows.

## 28.3.68 -lt Database Server Option

Sets the number of lock tables for each database.

### ↳ Syntax

```
start_iq -lt <n>
```

## Default

0

## Allowed Values

**0** The SAP IQ server/engine decides a suitable value to use.

**n** Any number that is a power of two between 1 and 128. The value is rounded up to the nearest power of two less than or equal to 128, if needed.

## Applies to

Applies to all operating systems.

## Remarks

At startup, SAP IQ selects how many lock tables to create based on how large the server appears to be. In general, more lock tables allow more simultaneous transactions, at a cost of slightly higher memory usage. On

large systems with hundreds of concurrent transactions, the selected number of lock tables may be too low. This option allows users to see how many lock tables were selected, to see if there are too few lock tables for a given workload, and to override the automatically selected number of lock tables if necessary.

## 28.3.69 -m Database Server Option

Truncates the transaction log when a checkpoint is done.

### ⌘ Syntax

```
start_iq -m ...
```

### Applies to

All operating systems.

### Remarks

This option truncates the transaction log when a checkpoint is done, either at shutdown or as a result of a checkpoint scheduled by the server.

#### ⚠ Caution

When this option is selected, there is no protection against media failure on the device that contains the database files.

This option provides a way to automatically limit the growth of the transaction log. Checkpoint frequency is still controlled by the `checkpoint_time` and `recovery_time` options (which you can also set on the command line).

The `-m` option is useful for limiting the size of the transaction log in situations where high volume transactions requiring fast response times are being processed, and the contents of the transaction log aren't being relied upon for recovery or replication. The `-m` option provides an alternative to operating without a transaction log at all, in which case a checkpoint would be required following each COMMIT and performance would suffer as a result. When the `-m` option is specified, there is no protection against media failure on the device that contains the database files. Other alternatives for managing the transaction log (for example, using the `BACKUP DATABASE` statement and events) should be considered before using the `-m` option.

To avoid database file fragmentation, place the transaction log on a separate device or partition from the database itself.

When this option is used, no operations can proceed while a checkpoint is in progress.

#### ⚠ Caution

Do not use the `-m` option with databases that are being replicated or synchronized.

## Related Information

[dbbackup Database Administration Utility \[page 19\]](#)

[dbfhide Database Administration Utility \[page 27\]](#)

## 28.3.70 -mp Database Server Option

Controls the number of heap memory pools used by the server catalog cache.

### ↳ Syntax

```
start_iq -mp <number>
```

## Default

4, or the number of logical processors in the system divided by 36 (whichever is higher)

## Applies To

All operating systems.

## Remarks

If your system has a large number of processors and is experiencing memory contention issues, try setting -mp to a value higher than the default. Check if the new -mp setting improves performance.

A value of 0 selects the default.

You can set to a value less than 4.

Server console messages display the -mp value as:

```
Number of heap pools: x
```

Usage messages display the -mp value as:

```
-mp <num> set number of memory pools
```

## 28.3.71 -n Database Server Option

Sets the name of the database server.

### ↔ Syntax

```
start_iq -n <server-name> <database-filename> ...
```

### Default

The name of the first database file (with the path and extension removed) that is started on the database server.

### Applies to

All operating systems.

### Remarks

When a database server starts, it attempts to become the default database server on that computer. The first database server to start when there is no default server becomes the default database server. Shared memory connection attempts on that computer that do not explicitly specify a database server name connect to the default server.

#### ⓘ Note

Use the `-xd` option for database servers being used by deployed applications, and that all clients explicitly specify the name of the database server to which they should connect by using the `ServerName (Server)` connection parameter. This ensures that the database connects to the correct database server when a computer is running multiple database servers.

There is no character set conversion performed on the server name. If the client character set and the database server character set are different, using extended characters in the server name can cause the server to not be found. If your clients and servers are running on different operating systems or locales, you should use 7-bit ASCII characters in the server name.

Database server names must be valid identifiers. Long database server names are truncated to different lengths depending on the protocol. Database server names cannot:

- begin with white space, single quotes, or double quotes
- end with white space
- contain semicolons, forward slashes (/), or backslashes (\)

- be longer than 250 bytes
- contain spaces when they are running on UNIX or Linux

#### Note

On Windows and UNIX/Linux, version 9.0.2 and earlier clients cannot connect to version 10.0.0 and later database servers with names longer than the following lengths:

- 40 bytes for Windows shared memory
- 31 bytes for UNIX or Linux shared memory
- 40 bytes for TCP/IP

The server name specifies the name to be used in the ServerName (Server) connection parameter of client application connection strings or profiles. With shared memory, unless -xd is specified, a default database server is used if a server name is not specified and there is at least one database server running on the computer.

Running multiple database servers with the same name is not recommended.

#### Note

There are two -n options. The -n option is positional. If it appears before any database file names, it is a server option and names the server. If it appears after a database file name, it is a database option and names the database.

## 28.3.72 -o Database Server Option

Prints all database server messages to the database server message log file.

#### Syntax

```
start_iq -o <filename> ...
```

### Applies to

All operating systems.

### Remarks

Print all database server messages, including informational messages, errors, warnings, and MESSAGE statement output, to the specified file, and to the database server messages window. If you specify the -qi option with -o, all messages appear only in the database server message log file.

Do not end the file name with `.log` because this can create problems for utilities that perform operations using the transaction log.

To find the name of the database server message log file, execute the following statement:

```
SELECT PROPERTY ( 'ConsoleLogFile' );
```

## 28.3.73 -oe Database Server Option

Specifies a file name to log startup errors, fatal errors, and assertions.

### ↳ Syntax

```
start_iq -oe <filename> ...
```

## Applies to

All operating systems.

## Remarks

Each line in the output log file is prefixed with the date and time. Startup errors include such errors as:

- Couldn't open/read database file: `<database-filename>`.
- A database server with that name has already started.

Do not end the file name with `.log` because this can create problems for utilities that perform operations using the transaction log.

## 28.3.74 -on Database Server Option

Specifies a maximum size for the database server message log, after which the file is renamed with the extension `.old` and a new file is started.

### ↳ Syntax

```
start_iq -on <size>[ k | m | g ] ...
```

## Applies to

All operating systems.

## Remarks

The `<size>` is the maximum file size for the database server message log, in bytes. Use `k`, `m`, or `g` to specify units of kilobytes, megabytes, or gigabytes respectively. The minimum size limit is 10 KB. By default, there is no maximum size limit.

When the database server message log reaches the specified size, the database server renames the file with the extension `.old`, and starts a new file with the original name.

### Note

If the `.old` database server message log file already exists, it is overwritten. To avoid losing old database server message log files, use the `-os` option instead.

This option cannot be used with the `-os` option.

Do not end the database server message log file name with `.log` because this can create problems for utilities that perform operations using the transaction log.

## 28.3.75 -os Database Server Option

Specifies a maximum size for the database server message log file, at which point the file is renamed.

### Syntax

```
start_iq -os <size>[ k | m | g ] ...
```

## Applies to

All operating systems.

## Remarks

The `<size>` is the maximum file size for logging database server messages, in bytes. Use `k`, `m`, or `g` to specify units of kilobytes, megabytes, or gigabytes respectively. The minimum size limit is 10 KB. By default, there is no maximum size limit.

Before the database server logs output messages to the database server message log file, it checks the current file size. If the log message will make the file size exceed the specified size, the database server renames the database server message log file to `yyymmddxx.slg`, where `<yyymmdd>` represents the year, month, and day the file was created, and `<xx>` is a number that starts at 00 and continues incrementing.

This option allows you to identify old database server message log files that can be deleted to free up disk space.

This option cannot be used with the `-on` option.

Do not end the database server message log file name with `.log` because this can create problems for utilities that perform operations using the transaction log.

## 28.3.76 `-ot` Database Server Option

Truncates the database server message log file and appends output messages to it.

### ↳ Syntax

```
start_iq -ot <logfile> ...
```

### Applies to

All operating systems.

### Remarks

The functionality is the same as the `-o` option except the database server message log file is truncated before any messages are written to it. To find the name of the database server message log file, execute the following statement:

```
SELECT PROPERTY ( 'ConsoleLogFile' );
```

Do not end the database server message log file name with `.log` because this can create problems for utilities that perform operations using the transaction log.

## 28.3.77 -p Database Server Option

Sets the maximum size of communication packets.

### ↳ Syntax

```
start_iq -p <integer> ...
```

### Default

7300 bytes (all operating systems)

### Applies to

All operating systems.

### Remarks

The minimum value is 1000 bytes and the maximum value is 65535.

You can change the communication buffer size for a connection by setting the CommBufferSize (CBSIZE) connection parameter.

## 28.3.78 -pc Database Server Option

Compresses all connections except for same-computer connections.

### ↳ Syntax

```
start_iq -pc ...
```

### Applies to

All operating systems.

## Remarks

The packets sent between a client and database server can be compressed using the `-pc` option. Compressing a connection may improve performance under some circumstances. Large data transfers with highly compressible data tend to get the best compression rates. This option can be overridden for a particular client by specifying `COMPRESS=NO` in the client's connection parameters.

By default, connections are not compressed. Specifying the `-pc` option compresses all connections except same-computer connections, web services connections, and TDS connections. TDS connections (including jConnect) do not support communication compression.

Same-computer connections over any communication link are not compressed, even if the `-pc` option or `COMPRESS=YES` connection parameter is used.

## 28.3.79 -phl Database Server Option

Sets history tracking for specified database server properties.

### ↔ Syntax

```
start_iq -phl { OFF | ON | NONE | <property-name>, ... }
```

On UNIX and Linux, quotation marks are required when more than one parameter is supplied:

## Default

ON

## Applies to

All operating systems.

## Allowed Values

**ON** When PropertyHistoryList is turned on, a default list of properties is tracked. The default is ON.

**OFF** When PropertyHistoryList is turned off property tracking is disabled for the database server.

**NONE** Setting this property to NONE enables property tracking, but no properties are tracked by the database server. Only properties requested by databases are tracked.

### Comma-delimited list of database server properties

Specify a comma-delimited list of database server properties to track.

## Remarks

The `-phl` database server option allows you to specify what database server properties, if any, are to be tracked by the server. If this option is not set or is set to ON, then the database server uses the default setting of the `PropertyHistoryList` database server property.

## 28.3.80 `-phs` Database Server Option

Sets the maximum amount of memory to use for tracking property history, in allotted time or bytes.

### Syntax

```
start_iq -phs { [<hh>:]<mm>:<ss> | <size> [ k | m ] | MAX | DEFAULT } ...
```

## Default

*DEFAULT*

## Applies to

All operating systems.

## Remarks

The `-phs` database server option specifies the maximum size allowed for tracking property history. The size is measured in allotted time per property or bytes of memory. The `<size>` parameter also increases the initial cache size of the database server to avoid having to grow the cache during start up. This increase in initial cache size prevents long start up times. If this option is not set, then the database server uses the *DEFAULT* setting of the `PropertyHistorySize` database server option.

If you specify a value for the `-phs` database server option, but the `-phl` database server option is set to OFF, then no property history is collected. If you set the `-phs` database server property to 0 or to a time-based value, then no space is allocated for property history tracking.

Specifying *MAX* sets the maximum size for tracking property history to 2% of the cache, or 256 MB, whichever is smaller.

Specifying *DEFAULT* sets the size for tracking property history to either 10 minutes or to a maximum of 2% of the cache.

### Note

If the specified size does not allow the database server to store the minimum amount of data for each tracked database server property (1024 bytes, or approximately 30 seconds), then an error is returned.

## 28.3.81 -pt Database Server Option

Increases or decreases the size limit at which TCP/IP packets are compressed.

### Syntax

```
start_iq -pt <size> ...
```

### Default

120 bytes

### Applies to

All operating systems.

### Remarks

This parameter takes an integer value representing the minimum byte-size of packets to be compressed. Values less than 80 are not recommended.

Under some circumstances, changing the compression threshold can help performance of a compressed connection by allowing you to compress packets only when compression will increase the speed at which the packets are transferred. The default setting should be appropriate for most cases.

If both client and server specify different compression threshold settings, the client setting applies.

## 28.3.82 -qi Database Server Option

Controls whether database server system tray icon and database server messages window appear.

### ↳ Syntax

```
start_iq -qi ...
```

### Applies to

Windows.

### Remarks

This option leaves no visual indication that the server is running, other than possible startup error windows. You can use either (or both) the -o or -oe log files to diagnose errors.

## 28.3.83 -qp Database Server Option

Specifies that messages about performance do not appear in the database server messages window.

### ↳ Syntax

```
start_iq -qp ...
```

### Applies to

All operating systems.

### Remarks

Do not display messages about performance in the database server messages window. Messages that are suppressed include the following:

- No unique index or primary key for table '`<table-name>`'
- Database file "`<mydatabase.db>`" consists of `<nnn>` fragments

## 28.3.84 -qs Database Server Option

Suppresses startup error windows.

### ↳ Syntax

```
start_iq -qs ...
```

### Applies to

Windows.

### Remarks

This option conceals startup error windows. Examples of startup errors include the database server not being able to open or read a database file or a database server not starting because another database server with the specified name is already running.

On Windows platforms, if the server is not being started automatically, these errors appear in a window and must be cleared before the server stops. These windows do not appear if the -qs option is used.

If there is an error loading the language DLL, the error is not logged to the -o or -oe logs, but rather to the Windows Application Event Log.

Usage errors are suppressed if -qs is on the command line, but not in @data expansion.

## 28.3.85 -qw Database Server Option

Specifies that the database server messages window does not appear.

### ↳ Syntax

```
start_iq -qw ...
```

### Applies to

All operating systems.

## Remarks

This option suppresses the database server messages window. On Windows platforms, the database server system tray icon is still visible. You can use either (or both) the `-o` or `-oe` log files to diagnose errors.

## 28.3.86 -s Database Server Option

Sets the user ID for Syslog messages.

### ⌘ Syntax

```
start_iq -s { none | user | daemon | local<n> } ...
```

## Applies to

UNIX and Linux.

## Remarks

Sets the system user ID used in messages to the Syslog facility. The default is `user` for database servers that are started in the foreground, and `daemon` for those that are run in the background (for example, started by `dbspawn`, started automatically by a client, or started with the `-ud` database server option).

A value of `none` prevents any Syslog messages from being logged. The `local<n>` argument allows you to use a facility identifier to redirect messages to a file. You can specify a number between 0 and 7, inclusive, for `<n>`. For more information, refer to the UNIX or Linux Syslog(3) man page.

## 28.3.87 -sb Database Server Option

Specifies how the database server reacts to broadcasts.

### ⌘ Syntax

```
start_iq -sb { 0 | 1 } ...
```

## Applies to

All operating systems. TCP/IP only.

## Remarks

The database server starts one or more UDP listeners so that it can respond to three types of broadcasts:

1. client connection broadcasts looking for this server.
2. database server enumeration broadcasts (such as those from the dblocate utility or the **Find Servers Wizard** of the administration tools **Connect** window).
3. broadcasts sent by another database server that is looking for any database servers with the same name.

For Embedded SQL connections, the dblocate utility and db\_locate\_servers function cause broadcast packets to be sent out on local networks in an attempt to find all database servers. The UDP listener within the database server then responds back to the sender with information about how to connect to the database server.

The -sb option controls the behavior of the UDP listeners within the database server.

When the -sb option is not specified, the database server responds to all three types of requests.

If you specify -sb 0, then the database server responds to (3) only. This forces clients to use a Host connection parameter or HOST protocol option when connecting to the database server. In addition if the server is using a port other than the default port (2638), then the client must also specify the server's port. Since the database server does not respond to database server enumeration broadcasts, the database server is not included in the output of dblocate.

If you specify -sb 1, then the database server responds to (1) and (3). Since the database server does not respond to database server enumeration broadcasts, the database server is not included in the output of dblocate.

## 28.3.88 -sf Database Server Option

Controls whether users have access to features for databases running on the current database server.

Unsecured features can be accessed by any user with the appropriate system role or privilege. A secured feature can only be accessed by a user that has been given the authorization to use a secured feature (by specifying an appropriate secured feature key) and who has the appropriate system role or privilege.

### ⌘ Syntax

```
start_iq -sf <feature-list> ...
```

```
<feature-list> :  
<feature-name> | <feature-set>[,[-]<feature-name> | <feature-set>] ...
```

## Parameters

### **none**

Specifies that no features are secured.

### **manage\_server**

This feature set prevents users from accessing all database server-related features. This set consists of the following features:

#### **processor\_affinity**

Prevents users from changing the processor affinity (the number of logical processors being used) of the database server.

**manage\_cockpitdb** Prevents users from enabling or disabling the Cockpit, or from changing the default file for the Cockpit. That is, `manage_cockpitdb` prevents users from executing `sa_server_option` with the CockpitDB option.

#### **manage\_listeners**

Prevents users from starting or stopping a connection listener by using the `sp_start_listener` or `sp_stop_listener` system procedure.

**manage\_property\_history** Prevents users from enabling and configuring the tracking of database server property values.

### **manage\_security**

This feature set prevents users from accessing features that allow the management of database server security. By default, these features are secured.

#### **manage\_features**

Prevents users from modifying the list of features that can be secured on the database server.

#### **manage\_keys**

Prevents the creation, modification, deletion, or listing of secured feature keys.

A user that has access to the `manage_keys` feature but not the `manage_features` feature cannot define a key with more features than those assigned to the user.

#### **manage\_disk\_sandbox**

Prevents users from temporarily changing disk sandbox settings by using the `sa_server_option` system procedure or the `sa_db_option` system procedure. The `manage_disk_sandbox` feature cannot be turned off for all databases or users. It can only be turned off for individual connections by using the `sp_use_secure_feature_key` system procedure.

### **server\_security**

This feature set prevents users from accessing features that can temporarily bypass security settings. By default, the following features, except for `trace_system_event`, are secured.

#### **disk\_sandbox**

Prevents users from performing read-write file operations on the database outside the directory where the main database file is located.

#### **trace\_system\_event**

Prevents users from creating user-defined trace events.

**database\_isolation** Allows database isolation to be temporarily turned off for the current connection.

## **all**

This feature set prevents users from accessing the following groups:

### **client**

This feature set prevents users from accessing all features that allow access to client-related input and output. This feature controls access to the client computing environment. This set consists of the following features:

#### **read\_client\_file**

Prevents the use of statements that can cause a client file to be read. For example, the READ\_CLIENT\_FILE function and the LOAD TABLE statement.

#### **write\_client\_file**

Prevents the use of all statements that can cause a client file to be written to. For example, the UNLOAD statement and the WRITE\_CLIENT\_FILE function.

### **remote**

This feature set prevents users from accessing all features that allow remote access or communication with remote processes. This set consists of the following features:

#### **remote\_data\_access**

Prevents the use of any remote data access services, such as proxy tables.

#### **send\_email**

Prevents the use of email system procedures, such as xp\_sendmail.

#### **send\_udp**

Prevents the ability to send UDP packets to a specified address by using the sa\_send\_udp system procedure.

#### **web\_service\_client**

Prevents the use of web service client stored procedure calls (stored procedures that issue HTTP requests).

### **local**

This feature set prevents users from accessing all local-related features. This feature controls access to the server computing environment. This set consists of the local\_call, local\_db, local\_io, and local\_log feature subsets.

#### **local\_call**

This feature set prevents users from accessing all features that can execute code that is not directly part of the database server and is not controlled by the database server. This set consists of the following features:

##### **cmdshell**

Prevents the use of the xp\_cmdshell procedure.

##### **external\_procedure**

Prevents the use of user-defined external stored procedures.

##### **external\_procedure\_v3**

See *SAP IQ Administration: User-Defined Functions*.

### **inmemory\_external\_procedure**

Disable the server's ability to make use of high performance in-process UDFs.

### **external\_library\_full\_text**

Prevents the use of a user-defined external term breaker library.

### **java**

Prevents the use of Java-related features, such as Java procedures.

## **local\_db**

This feature set prevents users from accessing all features related to database files. This set consists of the following features:

### **backup**

Prevents the use of the BACKUP DATABASE statement, and with it, the ability to run server-side backups. You can still perform client-side backups by using the dbbackup utility.

### **restore**

Prevents the use of the RESTORE DATABASE statement.

### **database**

Prevents the use of the CREATE DATABASE, ALTER DATABASE, and DROP DATABASE statements.

### **dbspace**

Prevents the use of the CREATE DBSPACE, ALTER DBSPACE, and DROP DBSPACE statements.

## **local\_env**

This feature set prevents users from accessing all features related to environment variables. This set consists of the following features:

### **getenv**

Prevents users from reading the value of any environment variable.

## **local\_io**

This feature set prevents users from accessing all features that allow direct access to files and their contents. This set consists of the following features:

### **create\_trace\_file**

Prevents the use of statements that create an event tracing target.

### **read\_file**

Prevents the use of statements that can cause a local file to be read. For example, the xp\_read\_file system procedure, the LOAD TABLE statement, and the use of OPENSTRING( FILE... ).

### **write\_file**

Prevents the use of all statements that can cause a local file to be written to. For example, the UNLOAD statement and the xp\_write\_file system procedure.

### **delete\_file**

Prevents the use of all statements that can cause a local file to be deleted. For example, securing this feature causes the dbbackup utility to fail if the -x or -xo options are specified.

#### **directory**

Prevents the use of directory class proxy tables. This feature is disabled when remote\_data\_access is disabled.

#### **file\_directory\_functions**

This feature set prevents users from accessing all of the following individual features:

##### **sp\_list\_directory**

Prevents the use of the sp\_list\_directory system procedure.

##### **sp\_create\_directory**

Prevents the use of the sp\_create\_directory system procedure.

##### **sp\_copy\_directory**

Prevents the use of the sp\_copy\_directory system procedure.

##### **sp\_move\_directory**

Prevents the use of the sp\_move\_directory system procedure.

##### **sp\_delete\_directory**

Prevents the use of the sp\_delete\_directory system procedure.

##### **sp\_copy\_file**

Prevents the use of the sp\_copy\_file system procedure.

##### **sp\_move\_file**

Prevents the use of the sp\_move\_file system procedure.

##### **sp\_delete\_file**

Prevents the use of the sp\_delete\_file system procedure.

##### **sp\_disk\_info**

Prevents the use of the sp\_disk\_info system procedure.

#### **local\_log**

Prevents users from accessing all logging features that result in creating or writing data directly to a file on disk. This set consists of the following features:

##### **request\_log**

Prevents the ability to change the request log file name and also prevents the ability to increase the limits of the request log file size or number of files. You can specify the request log file and limits on this file in the command to start the database server; however, they cannot be changed once the database server is started. When request log features are disabled, you can still turn request logging on and off and reduce the maximum file size and number of request logging files.

##### **console\_log**

Prevents the ability to change the database server message log file name using the ConsoleLogFile option of the sa\_server\_option system procedure. Securing this feature also prevents the ability to increase the maximum size of the database server message log file

using the ConsoleLogMaxSize option of the sa\_server\_option system procedure. You can specify a server log file and its size when starting the database server.

### **webclient\_log**

Prevents the ability to change the web service client log file name using the WebClientLogFile option of the sa\_server\_option system procedure. You can specify a web service client log file when starting the database server.

## **Default**

none

## **Applies to**

All operating systems.

## **Remarks**

This option allows the owner of the database server to control whether users have access to features for databases running on the database server. The -sk database server option allows the owner of the database server to create the SYSTEM secured feature key that permits users access to features secured by the -sf database server option.

If you start a database server without specifying the -sk database sever option, the features specified by the -sf database sever option and some default features are secured and there is no way to change which features are secured while the database server is running. You cannot create the SYSTEM secure feature key later using a system stored procedure. You must shut down the database server and specify the -sk database sever option when you restart it.

The <feature-list> is a comma-separated list of feature names or feature sets to secure for the database server. Securing a feature makes it inaccessible to all database users other than administrators. Specifying a feature set secures all the features included in the set. To secure one or more, but not all, of the features in the feature set, specify the individual feature name.

Use <feature-name> to indicate that the feature should be secured (made inaccessible), and -<feature-name> or <feature-name>- to indicate that the feature should be unsecured (accessible to all database users). For example, the following command indicates that only dbspace features are accessible to all users:

### **Note**

Sub-features of feature sets that are secured by default, cannot be unsecured from the command line. In other words the following command will not work:

```
-sf manage_security,-manage_keys
```

To use sub-features of feature sets that are secured by default, call the `sp_use_secure_feature_key` system procedure specifying the SYSTEM secure feature key which includes `MANAGE_KEYS` and the password specified by the `-sk` option :

```
CALL sp_use_secure_feature_key( 'system' , 'letmeinweyou' );
```

## Example

The following command starts a database server named `secure_server` with all local data access features secured, except for the `dbspace` feature.

```
start_iq -n secure_server -sf all,-dbspace
```

The following command starts a database server named `secure_server` with all remote data access features secured, except for the `web_service_client` feature. The key specified by the `-sk` option can be used later with the `sp_use_secure_feature_key` system procedure to make these features accessible to all users on the current connection.

```
start_iq -n secure_server -sf remote,-web_service_client -sk j978kls12
```

If a user connected to a database running on the `secure_server` database server uses the `sp_use_secure_feature_key` system procedure with the `authorization_key` parameter set to the same value as that specified by `-sk`, that connection has access to the remote data access features:

```
CALL sp_use_secure_feature_key ( 'MyKey' , 'j978kls12' );
```

The following command secures all features, with the exception of local database features:

```
start_iq -n secure_server -sf all,-local_db
```

## 28.3.89 -sk Database Server Option

Creates the SYSTEM secured feature key and sets the authorization code for it. This permits access to features that are secured for the database server.

### ↔ Syntax

```
start_iq -sk <auth_code> ...
```

## Applies to

All operating systems.

## Remarks

When you secure features for a database server by using the `-sf` option, you can also include the `-sk` option, which creates the SYSTEM secured feature key and sets the authorization code for it. This authorization code can be used with the `sp_use_secure_feature_key` system procedure to allow access to secured features for a connection. That connection can also use the `sa_server_option` system procedure to modify the features or feature sets that are secured for all databases running on the database server.

The authorization code must be a non-empty string of at least six characters, and it cannot contain double quotes, control characters (any character less than 0x20), or backslashes..

If the authorization code specified with the `sp_use_secure_feature_key` system procedure does not match the value specified by `-sk`, no error is given and the features specified by `-sf` remain secured for the connection.

If you specify `-sk` without `-sf`, only default features like those contained in the `manage_security` feature set are secured, but you can use the SYSTEM secured feature key while the database server is running to gain access to and change the secured feature settings.

## Example

The following command starts a database server named `secure_server` with the backup feature and some default features secured. The authorization code specified by the `-sk` option is used later to access these features for a specific connection.

```
start_iq -n secure_server -sf backup -sk j978kls12
```

You use the `sp_use_secure_feature_key` system procedure, specifying SYSTEM for the key name and the authorization code that was specified with the `-sk` option. This allows you to perform backups, change the features that are secured on the `secure_server` database server, and create other keys.

```
CALL sp_use_secure_feature_key( 'SYSTEM' , 'j978kls12' );
```

You can secure all features for databases running on `secure_server` by executing the following statement:

```
CALL sa_server_option( 'SecureFeatures', 'all' );
```

You can create other keys to permit access to selected features by other users (provided you disclose the key name and authorization code).

```
CALL sp_create_secure_feature_key ( 'client_access' , 'client_auth_code' ,  
'client' );
```

## 28.3.90 -su Database Server Option

Sets the user ID and password for the utility database (utility\_db), or disables connections to the utility database.

### Syntax

```
start_iq -su { <user-ID>,<password> | <password> | NONE } ...
```

### Applies to

All operating systems.

### Remarks

This option specifies the initial password, and optionally the user ID, for the utility database. The minimum length of the password is always 6, is case sensitive, and cannot contain a comma. Specify *NONE* instead of a user ID and/or password to disable all connections to the utility database. If you do not specify a *<user-ID>*, then the database server uses the default value DBA.

- If you are using a network database server and do not specify the -su database server option, then connections to the utility database are not allowed.
- Allowing connections to the utility database for a network database server is useful for cases when the database server is running, but it is not possible to connect to the database.

You can execute a CREATE USER *<user-ID>* IDENTIFIED BY *<new-password>* statement while connected to utility\_db to change the password for the user *<user-ID>* of the utility database. The REVOKE CONNECT FROM *<user-ID>* statement can be used to disable connections to the utility\_db database. Not all SQL statements are supported for the utility database.

To avoid having the utility database password in clear text on the command line, you can use the dbfhide utility to encrypt a file containing *<password>* and then reference the encrypted file on the command line.

### Example

The following command disables all connections to the utility database:

```
start_iq -su none c:\inventory.db
```

In the following example, the file named util\_db\_uid\_pwd.cfg that contains the utility database user ID and password is encrypted by using dbfhide and renamed util\_db\_uid\_pwd\_hide.cfg:

```
dbfhide util_db_uid_pwd.cfg util_db_uid_pwd_hide.cfg
```

The `util_db_pwd_hide.cfg` file can then be used to specify the utility database user ID and password:

```
start_iq -su @util_db_uid_pwd_hide.cfg -n my_server c:\inventory.db
```

## 28.3.91 -tdsl Database Server Option

Sets the TDS login mode.

### ↔ Syntax

```
start_iq -tdsl { ALL | RSA | RSANONCE } ...
```

### Default

*ALL*

### Applies to

All operating systems.

### Remarks

This option restricts the type of TDS login requests that the database server supports.

Mode type	Description
ALL	Encrypted and unencrypted passwords are allowed (including those with a nonce) in TDS login requests:
RSA	Only RSA encrypted passwords (including those with a nonce) are allowed in TDS login requests:
RSANONCE	Only RSA encrypted passwords with a nonce are allowed in TDS login requests:

When you make login requests from a TDS application that supports RSA without a nonce, the database server generates a new set of encryption key for the login requests. Generating new encryption keys can be time-consuming.

When you make login requests from a TDS application that supports RSA with a nonce, the database server reuses a set of RSA encryption keys. These encryption keys are re-generated every 24 hours. By reusing the RSA encryption keys, performance can improve while protecting the database server from replay attacks.

Both jConnect and Open Client support RSA login requests with and without a nonce.

## 28.3.92 -ti Database Server Option

Disconnects inactive connections.

### ↳ Syntax

```
start_iq -ti <minutes> ...
```

### Default

240 (4 hours)

### Applies to

All operating systems.

### Remarks

Disconnects connections that haven't submitted a request for the specified number of <minutes>. The maximum value is 32767. A client computer in the middle of a database transaction holds locks until the transaction is ended or the connection is disconnected. The -ti option is provided to disconnect inactive connections, freeing their locks.

Setting the value to zero disables checking of inactive connections, so that no connections are disconnected. If the IdleTimeout (IDLE) connection parameter is not used, then the idle timeout value for TCP/IP connections is controlled by the -ti database server option. If both the -ti database server option and the IdleTimeout (IDLE) connection parameter are specified, then the idle timeout value is controlled by the connection parameter.

Idle timeout is not supported for TDS clients, including jConnect clients.

## 28.3.93 -tl Database Server Option

Sets the period at which to send liveness packets.

### ↳ Syntax

```
start_iq -tl <seconds> ...
```

### Applies to

All operating systems. TCP/IP only.

### Remarks

A liveness packet is sent periodically across a client/server TCP/IP communications protocol to confirm that a connection is intact. If the server runs for a LivenessTimeout period (default 2 minutes) without detecting a liveness packet on a connection, the communication is severed, and the server drops the connection associated with that client. UNIX and Linux non-threaded clients and TDS connections do not do liveness checking.

The -tl option on the server sets the LivenessTimeout value for all clients that do not specify a liveness period.

Liveness packets are sent when a connection hasn't sent any packets for between one third and two thirds of the LivenessTimeout value.

When there are more than 200 connections, the server automatically calculates a higher LivenessTimeout value based on the stated LivenessTimeout value, so the server can handle a large number of connections more efficiently. Liveness packets are sent between one third and two thirds of the LivenessTimeout on each idle connection. Large numbers of liveness packets aren't sent at the same time. Liveness packets that take a long time to send could be sent after two thirds of the LivenessTimeout. A warning appears in the database server message log if the liveness sends take a long time. If this warning occurs, consider increasing the LivenessTimeout value.

Although it isn't generally recommended, you can disable liveness by specifying the following:

```
start_iq -tl 0 -n my_server
```

Rather than disabling the LivenessTimeout option, consider increasing the value to one hour as follows:

```
start_iq -tl 3600 -n my_server
```

## 28.3.94 -tmf Database Server Option

Forces transaction manager recovery for distributed transactions.

### ↔ Syntax

```
start_iq -tmf ...
```

### Applies to

Windows.

### Remarks

Used during recovery of distributed transactions when the distributed transaction coordinator isn't available. If DTC cannot be located, the outstanding operations are rolled back and recovery continues. It can also be used when starting a database with distributed transactions in the transaction log on a platform where the DTC isn't available.

### ⚠ Caution

If you use this option, distributed transactions are not recovered properly. It is not intended for routine use.

## 28.3.95 -tq Database Server Option

Shuts down the server at a specified time.

### ↔ Syntax

```
start_iq -tq { <datetime> | <time> } ...
```

### Applies to

All operating systems.

## Remarks

This option is useful for setting up automatic off-line backup procedures.

The format for the time is in `<hh>:<mm>` (24 hour clock), and can be preceded by an optional date. If a date is specified, the date and time must be enclosed in double quotes and be in the format `<YYYY/MM/DD><HH>:<MM>`.

## 28.3.96 -u Database Server Option

Opens files using the operating system disk cache.

### ⌘ Syntax

```
start_iq -u ...
```

## Applies to

All operating systems.

## Remarks

Files are opened using the operating system disk cache in addition to the database cache.

While the operating system disk cache may improve performance, better performance is often obtained by using only the database cache.

If the server is running on a dedicated computer, you shouldn't use the -u option, as the database cache itself is generally more efficient. You may want to use the -u option if the server is running on a computer with several other applications (so that a large database cache may interfere with other applications) and yet IO-intensive tasks are run intermittently on the server (so that a large cache will improve performance).

## 28.3.97 -ud Database Server Option

(UNIX servers) Causes the process to run as a daemon in the root directory. SAP recommends that you do not use this switch in IQ servers.

## 28.3.98 -uf Database Server Option

Specifies the action that the database server takes when a fatal error or assertion failure occurs on a database server.

### ↳ Syntax

```
start_iq -uf <action> ...
```

### Default

default (for UNIX and Linux)

defunct (for Windows)

### Allowed Values

#### **abort**

The database server is shut down and a core file is generated.

#### **default**

The database server behaves in the same manner as abort, except when a device-full fatal error occurs. In this case, it behaves in the same manner as defunct. This action prevents the system from trying to write a core file on a full device.

#### **defunct**

The database server continues running and does not call abort. Any new connection attempts that are made to the database server receive the original error.

### Applies to

All operating systems.

## 28.3.99 -ufd Database Server Option

Specifies the action that the database server takes when a fatal error or assertion failure occurs on a database.

### ↳ Syntax

```
start_iq -ufd <action> ...
```

## Default

abort

## Allowed Values

### **abort**

The affected database is stopped. The status of the database server and other databases remains unchanged.

### **escalate**

The database assertion failure becomes a database server assertion failure and is handled according to the setting for the `-uf` database server option.

### **restart**

This value is deprecated; use `restart_escalate` instead.

### **restart\_abort**

The database server automatically restarts the database. A restart is considered unsuccessful if another database assertion failure occurs within 60 seconds of the restart. If the second consecutive restart is unsuccessful, the affected database is stopped. The status of the database server and other databases remains unchanged.

### **restart\_escalate**

The database server automatically restarts the database. A restart is considered unsuccessful if another database assertion failure occurs within 60 seconds of the restart. If the second consecutive restart is unsuccessful, then the database assertion failure causing the unsuccessful restart becomes a server assertion failure and is handled according to the setting for the `-uf` database server option.

## Applies to

All operating systems.

## Remarks

The `-ufd` database server option must precede any database file specification.

## 28.3.100 -ut Database Server Option

(UNIX servers) Causes the server to touch catalog store temporary files at intervals specified by `<min>`.

### ↳ Syntax

```
start_iq -ut <min>
```

## 28.3.101 -v Database Server Option

Displays the database server version in a message box (Windows) or in a version string (UNIX/Linux).

### ↳ Syntax

```
start_iq { -v | -v2 }
```

## 28.3.102 -x Database Server Option

Specifies server-side network communications protocols.

### ↳ Syntax

```
start_iq -x { ALL | NONE | TCPIP [ ( <parm>=<value>;... ) ] ... }
```

## Allowed Values

Regardless of which settings you choose for the `-x` option, the database server always accepts shared memory connections.

By default, the network database server starts the shared memory and TCP/IP protocols.

You can specify the following values for the `-x` option:

### ALL

Listen for connection attempts by the client using shared memory and TCP/IP protocols.

### NONE

Listen for connection attempts by the client using only the shared memory protocol.

### TCPIP (TCP)

Listen for connection attempts by the client using the shared memory and TCP/IP protocols.

## Applies to

All operating systems.

## Remarks

Use the `-x` option to specify which communications protocols you want to use to listen for client connections.

### Note

Use the `-ec` database server option to enable transport layer security (TLS), or simple obfuscation, to encode communication protocol packets (such as DBLIB and ODBC) transmitted to and from all clients. See [-ec Database Server Option \[page 123\]](#).

For the TCP/IP protocol, additional parameters may be provided, in the following format:

```
-x tcpip(<parm1>=<value1>;<parm2>=<value2>;...)
```

For UNIX and Linux, quotation marks are required if more than one parameter is supplied or certain punctuation characters are used, such as parentheses and semicolons. For example:

```
-x "tcpip(<parm1>=<value1>;<parm2>=<value2>;...)"
```

When the database server listens for TCP/IP connections, by default it listens to all network adapters on port 2638.

## Example

Start a secure server:

```
start_iq -x tcpip(port=1234;TDS=ENCRYPTED) -ec  
"TLS(identity=myserver.id;identity_password=XXXXXX)"
```

## Example

Allow shared memory and TCP/IP connections on the network server:

```
start_iq -n server_name "%IQDIRSAMP%\iqdemo.db"
```

Use the `-x` option to specify protocol options to tune the behavior of TCP/IP. Allow the network server to use two specific network cards:

```
start_iq -x "tcpip(MyIP=192.75.209.12,192.75.209.32)" "%IQDIRSAMP%\iqdemo.db"
```

## Related Information

[-ec Database Server Option \[page 123\]](#)

[Network Communications Parameters](#)

[TDS Communication Parameter](#)

### 28.3.103 -xd Database Server Option

Prevents the database server from becoming the default database server.

#### ⌘ Syntax

```
start_iq -xd
```

#### Remarks

When a database server starts, it attempts to become the default database server on that computer. The first database server to start when there is no default server becomes the default database server. Shared memory connection attempts on that computer that do not explicitly specify a database server name connect to the default server.

Specifying this option prevents the database server from becoming the default database server. If this option is specified, clients that do not specify a database server name cannot find the database server over shared memory. The `-xd` option also prevents the database server from using the default TCP port. If a TCP port is not specified, the database server uses a port other than port 2638.

### 28.3.104 -xs Database Server Option

Specifies server-side Web services communications protocols.

#### ⌘ Syntax

```
start_iq -xs { NONE  
| HTTP [ ( <parm>=<value>;... ) ]  
| HTTPS [ ( <parm>=<value>;... ) ] } ...
```

On UNIX and Linux, quotation marks are required if more than one parameter is supplied.

## Allowed Values

You can specify any of the following:

### HTTP

Listen for web requests by the client using the HTTP protocol. The default port on which to listen is 80.

### HTTPS

Listen for web requests by the client using the HTTPS protocol. The default port on which to listen is 443. Specify the server's certificate and password to use HTTPS. The certificate must be an RSA certificate because HTTPS uses RSA encryption.

The HTTP server supports HTTPS connections using TLS version [1.0](#), [1.1](#), [1.2](#), or [1.3](#). TLS 1.2 and 1.3 are selected by default with the server and client negotiating the highest protocol they both support. The `min_tls_version` network protocol option can be used to change allowed protocols.

Specify HTTPS, or HTTPS with FIPS=Y for FIPS-certified RSA encryption. FIPS-certified HTTPS uses a separate certified library, but is compatible with HTTPS.

### NONE

(Default) Don't listen for web requests.

### ODATA [Deprecated]

Listen for web requests by the client using the OData protocol. Specifying `-xs ODATA` starts an OData server that runs as a subprocess of the database server. For more information about OData protocol options, see the *Network protocol options* topic.

#### Note

Specifying `-xs ODATA` on platforms other than Windows or Linux (not ARM) returns an error.

### parm

A network protocol option.

### odata-parm

An OData protocol option.

## Applies To

All operating systems.

## Remarks

To fully secure the server, clients and server should mutually trust each others' certificate chains and ensure that only secure communication is accepted by the server. Clients should perform checks on the server's certificate, via the host name check or the name/company/unit checks. Self-signed server and client certificates are not recommended. See *Using Client-side Certificates for TLS* for more details on how to setup mutual trust.

Use the `-xs` option to specify which web protocols you want to use to listen for requests.

If you don't specify the `-xs` option, then the database server doesn't attempt to listen for web requests.

To specify multiple protocols, specify the `-xs` option for each protocol or specify multiple protocols with one `-xs` database server option. The database server listens for web requests using all the specified protocols.

If you specify both the HTTP and OData protocols, then at least one of them must specify a nondefault HTTP port number. If you specify both the HTTPS and OData protocols, then at least one of them must specify a nondefault HTTPS port number.

### Note

To start multiple web servers at the same time, change the port for one of them since they all have the same default port.

Use the HTTPS and FIPS-validated HTTPS protocol for transport layer security.

## Example

Listen for HTTP web requests on port 80:

```
start_iq web.db -xs HTTP(PORT=80)
```

Listen for web requests using HTTPS:

```
start_iq web.db -xs  
HTTPS(FIPS=N;PORT=82;IDENTITY=ecserver.id;IDENTITY_PASSWORD=test)
```

Listen for web requests using HTTPS with the minimum TLS version set to 1.3 to avoid weaker TLS 1.0, 1.1, and 1.2:

```
start_iq web.db -xs  
HTTPS(FIPS=N;PORT=82;IDENTITY=ecserver.id;IDENTITY_PASSWORD=test;MIN_TLS_VERSION=  
1.3)
```

Listen for web requests using HTTPS with the minimum TLS version set to 1.2, and with ciphers that have perfect forward secrecy (PFS) strength or SHA384 hashing:

```
start_iq web.db -xs  
HTTPS(FIPS=N;PORT=82;IDENTITY=ecserver.id;IDENTITY_PASSWORD=test;MIN_TLS_VERSION=  
1.2;SET_CIPHERS=PFS:mSHA384)
```

Listen on ports 80 and 8080:

```
start_iq -xs HTTP(port=80),HTTP(PORT=8080)
```

or

```
start_iq -xs HTTP(port=80) -xs HTTP(PORT=8080)
```

[Deprecated] Start a database server that listens on port 8080 for OData requests:

```
start_iq -xs odata(ServerPort=8080) c:\mydatabase.db
```

[Deprecated] Run the following command to start a database server that listens on port 8443 for secure OData requests using the key store `c:\mykeys.jks` and the key store password `password`, and specifying that the database server does use the default port to listen for nonsecure OData requests:

```
start_iq -xs
odata(SecureServerPort=8443;SSLKeyStore=c:\mykeys.jks;SSLKeyStorePassword=password;ServerPort=)
```

See *Using Client-side Certificates for TLS* for examples using `trusted_certificate` on the server side.

## Related Information

[Network Communications Parameters](#)  
[Using Client-side Certificates for TLS](#)

## 28.3.105 -z Database Server Option

Displays diagnostic communication messages, and other messages, for troubleshooting purposes.

### ↳ Syntax

```
start_iq -z ...
```

## Applies to

All operating systems.

## Remarks

This should only be used when tracking problems. The information appears in the database server messages window.

## 28.3.106 -ze Database Server Option

Displays database server environment variables in the database server messages window.

### ↳ Syntax

```
start_iq -ze ...
```

## Applies to

All operating systems.

## Remarks

When you specify the `-ze` option, environment variables are listed in the database server messages window on startup. You can log the contents of the database server messages window to a file by specifying the `-o` option when starting the database server.

## Example

The following command starts a database server named `myserver`, and outputs the environment variables set for the server to the database server messages window and the file `server-log.txt`.

```
start_iq -n myserver -ze -o server-log.txt
```

## 28.3.107 -zl Database Server Option

Turns on capturing of the most recently prepared SQL statement for each connection to databases on the server.

### ⌘ Syntax

```
start_iq -zl ...
```

## Applies to

All operating systems.

## Remarks

This feature can also be turned on using the `RememberLastStatement` server setting. You can obtain the most recently prepared SQL statement for a connection using the `LastStatement` value of the `CONNECTION_PROPERTY` function. The `sa_conn_activity` stored procedure allows you to obtain the most recently prepared SQL statement for all current connections to databases on the server.

The LastStatement value is set when a statement is prepared, and is cleared when a statement is dropped. Only one statement string is remembered for each connection.

If sa\_conn\_activity reports a non-empty value for a connection, it is most likely the statement that the connection is currently executing. If the statement had completed, it would likely have been dropped and the property value would have been cleared. If an application prepares multiple statements and retains their statement handles, the LastStatement value does not reflect what a connection is currently doing.

For stored procedure calls, only the outermost procedure call appears, not the statements within the procedure.

#### ⚠ Caution

When -zl is specified or when the RememberLastStatement server setting is turned on, any user can call the sa\_conn\_activity system procedure or obtain the value of the LastStatement connection property to find out the most recently prepared SQL statement for any other user. The user must have either the SERVER OPERATOR, MONITOR, or DROP CONNECTION system privilege.

## 28.3.108 -zn Database Server Option

Specifies the number of request log file copies to retain.

#### ↗ Syntax

```
start_iq -zn <integer> ...
```

### Applies to

All operating systems.

### Remarks

If request logging is enabled over a long period of time, the request log file can become large. The -zn option allows you to specify the number of request log file copies to retain. It only takes effect if -zs is also specified. The -zs option allows you to create a new log file and rename the original log file when the original log file reaches a specified size.

For example, if you redirect request logging information to the file req.out, and specify five request log file copies using the -zn option, the server creates files in the following order: req.out.1, req.out.2, req.out.3, req.out.4, and req.out.5. When these files exist and the active request log fills again, the following happens:

- req.out.1 is deleted
- the files req.out.2 to req.out.5 are renamed req.out.1 to req.out.4

- the copy of the active log is renamed `req.out.5`

Request logging is turned on using the `-zr` option and redirected to a separate file using the `-zo` option. You can also set the number of request logs using the `sa_server_option` system procedure where `<nn>` specifies the number of request log file copies:

```
CALL sa_server_option('RequestLogNumFiles', <nn>);
```

## Example

In the following example, request logging information is output to a request log file named `mydatabase.log`, which has a maximum size of 10 KB, and three copies of the request log are kept:

```
start_iq "c:\my data\mydatabase.db" -zr all -zn 3  
-zs 10 -zo mydatabase.log
```

## 28.3.109 -zo Database Server Option

Redirects request logging information to a file separate from the regular log file.

### ↔ Syntax

```
start_iq -zo <filename> ...
```

## Applies to

All operating systems.

## Remarks

Request logging is turned on using the `-zr` option. You can direct the output from this file to a different file that is not the regular log file by specifying the `-zo` option.

This option also prevents request logging from appearing in the database server messages window.

## 28.3.110 -zoc Database Server Option

Redirects web service client information to a file.

### ↔ Syntax

```
start_iq -zoc <filename> ...
```

### Applies to

All operating systems.

### Remarks

The web service client log file contains HTTP requests and transport data recorded for outbound web service client calls. The web service client log file can also be specified using the `sa_server_option` system procedure:

```
CALL sa_server_option( 'WebClientLogFile', 'clientinfo.txt' );
```

Logging is enabled automatically when you specify the `-zoc` server option. You can enable and disable logging to this file using the `sa_server_option` system procedure:

```
CALL sa_server_option( 'WebClientLogging', 'ON' );
```

### Example

The following command starts the database server so that it listens for HTTP web requests on port 80, and logs outbound web service client information to the file `clientinfo.txt`:

```
start_iq web.db -xs HTTP(PORT=80) -zoc clientinfo.txt
```

## 28.3.111 -zr Database Server Option

Enables request logging of operations.

### ↔ Syntax

```
start_iq -zr { SQL | HOSTVARS | PLAN | PROCEDURES | TRIGGERS | OTHER |  
BLOCKS | REPLACE | ALL | NONE } ...
```

## Allowed Values

### SQL

enables logging of the following:

- START DATABASE statements
- STOP DATABASE statements
- STOP SERVER statements
- Statement preparation and execution
- EXECUTE IMMEDIATE statement
- Option settings
- COMMIT statements
- ROLLBACK statements
- PREPARE TO COMMIT operations
- Connects and disconnects
- Beginnings of transactions
- DROP STATEMENT statements
- Cursor explanations
- Cursor open, close, and resume
- Errors

### HOSTVARS

enables logging of host variable values. If you specify HOSTVARS, the information listed for SQL is also logged.

### PLAN

enables logging of execution plans (short form). Execution plans for procedures are also recorded if logging of procedures (PROCEDURES) is enabled.

### PROCEDURES

enables logging of statements executed from within procedures and user-defined functions.

### TRIGGERS

enables logging of statements executed from within triggers.

### OTHER

enables logging of additional request types not included by SQL, such as FETCH and PREFETCH. However, if you specify OTHER but do not specify SQL, it is the equivalent of specifying SQL+OTHER. Including OTHER can cause the request log file to grow rapidly and could negatively affect server performance.

### BLOCKS

enables logging of details showing when a connection is blocked and unblocked on another connection.

### REPLACE

at the start of logging, the existing request log is replaced with a new (empty) one of the same name. Otherwise, the existing request log is opened and new entries are appended to the end of the file.

### ALL

logs all supported information. This setting is equivalent to specifying SQL+PLAN+HOSTVARS+PROCEDURES+TRIGGERS+OTHER+BLOCKS. This setting can cause the request log file to grow rapidly and could negatively affect server performance.

#### **NO or NONE**

turns off logging to the request log.

## **Applies to**

All operating systems.

## **Remarks**

This option should only be used when tracking problems. The information appears in the database server messages window or is sent to the request log. When you specify multiple values, they are separated with a , or a +.

Once the database server is started, you can change the request log settings to log more or less information using the sa\_server\_option system procedure.

### **Note**

The -zo database server option is recommended whenever using the -zr database server option. Using the -zr database server option without using the -zo database server option can impact database server performance and in some cases increase memory usage.

You can find the current value of the RequestLogging setting using the following query:

```
SELECT PROPERTY( 'RequestLogging' );
```

## **28.3.112 -zs Database Server Option**

Limits the size of the request log.

### **Syntax**

```
start_iq -zs <size>[ k | m | g ] ...
```

## **Applies to**

All operating systems.

## Remarks

Request logging is turned on using the `-zr` option, and redirected to a separate file using the `-zo` option. You can limit the size of the file using the `-zs` option.

The `<size>` is the maximum file size for the request log, in bytes. Use `k`, `m`, or `g` to specify units of kilobytes, megabytes, or gigabytes respectively.

If you specify `-zs 0`, then there is no maximum size for the request logging file, and the file is never renamed. This is the default value.

When the request log file reaches the size specified by either the `-zs` option or the `sa_server_option` system procedure, the file is renamed with the extension `.old` appended (replacing an existing file with the same name if one exists). The request log file is then restarted.

## Example

The following example shows how the `-zs` option is used to control log file size. Suppose you start a database server with the following command line:

```
start_iq -zr all -zs 10k -zo mydatabase.log
```

A new log file `mydatabase.log` is created. When this file reaches 10 KB in size, any existing `mydatabase.old` files are deleted, `mydatabase.log` is renamed to `mydatabase.old`, and a new `mydatabase.log` file is started. This process is repeated each time the `mydatabase.log` file reaches the specified size (in this case 10 KB).

## 28.4 start\_iq Database File Parameters

This topic lists the parameters of the database server/database file. You specify the database file after the server options in the command syntax.

start_iq Database file parameter	Description
<code>-n &lt;server-name&gt;</code>	Specifies the name of the database server.

**start\_iq Database file parameter**

**Description**

`database-file` Specifies the database file name. If `<database-file>` is specified without a file extension, SAP IQ looks for `<database-file>` with extension `.db`.

If you use a relative path, the path is read relative to the current working directory of the server. You can supply a full path.

On Windows you can supply a path that conforms to the Universal Naming Convention (UNC) format:

```
\\server\volume\path\file.ext
```

**⚠ Caution**

The database file must be on the same machine as the database server. Managing a database file that is located on a network drive can lead to file corruption.

## 28.5 start\_iq Database Options

Available switches for the `start_iq <database-options>` parameters.

Specify these options after the database file. These options apply only to that database. These options apply only to the preceding database in the command syntax.

There are two forms of syntax for `start_iq` database options:

- When specifying options in a configuration file, do not enclose the option value in quotation marks. For example:

```
iqdemo.db -ek xxx
```

- When specifying options on the command line, enclose the option value in quotation marks. For example:

```
start_iq @iqdemo.cfg iqdemo.db -ek 'xxx'
```

**In this section:**

[-a Database Option \[page 210\]](#)

Applies the named transaction log.

[-ad Database Option \[page 211\]](#)

Specifies the directories containing transaction log files to be applied to the database.

[-al Database Option \[page 212\]](#)

Extends LOGIN\_MODE for LDAPUA only to a select number of users using Standard authentication.

[-ar Database Option \[page 213\]](#)

Specifies that any transaction log files located in the same directory as the current transaction log should be applied to the database.

[-as Database Option \[page 214\]](#)

Specifies that the database should continue to run after transaction logs have been applied (used in conjunction with -ad or -ar).

[-dh Database Option \[page 215\]](#)

Makes a database undetectable when the Server Location utility `dblocate -d` is run against the server.

[-ds Database Option \[page 215\]](#)

Specifies the directory where the dbspaces for the database and the transaction log are located.

[-ek Database Option \[page 216\]](#)

Specifies the database encryption key.

[-iqfreq Database Option \[page 216\]](#)

Marks the specified database as in use and restores the IQ main store portion of the database to its last known consistent state.

[-m Database Option \[page 217\]](#)

A server startup option that truncates (deletes) the transaction log when a checkpoint is done, either at shutdown or as a result of a checkpoint scheduled by the server.

[-n Database Option \[page 218\]](#)

Provides an alternate name, or nickname, for the database.

[-r Database Option \[page 218\]](#)

Forces all databases that start on the database server to be read-only.

[-sm Database Option \[page 219\]](#)

Provides an alternate database server name that can be used to access the read-only mirror database.

[-sn Database Option \[page 219\]](#)

Provides an alternate server name for a single database running on a database server.

[-xp Database Option \[page 219\]](#)

Provides information to an operational server that allows it to connect to its partner and to the arbiter when database mirroring is being used.

## Related Information

[dblocate Database Administration Utility \[page 33\]](#)

### 28.5.1 -a Database Option

Applies the named transaction log.

#### Syntax

```
start_iq [ <server-options> ] <database-file> -a <log-filename> ...
```

## Applies to

All operating systems.

## Remarks

This option is used to recover from media failure on the database file. When this option is specified, the database server applies the log and then shuts down. It doesn't continue to run. If you need to apply multiple transaction logs, you must know the correct order in which to apply them when using `-a`. The database server automatically applies multiple transaction logs in the correct order if you use the `-ad` or `-ar` option.

If the specified log file name uses a relative path, then this directory is read relative to the database file directory.

The `-a` database option must be specified after the `<database-file>`, and applies only to that database.

Specifying a cache size when starting the server can reduce recovery time.

## Example

The following example applies the transaction log file `mydemo.log` to a backup copy of the `mydemo` database.

```
start_iq "c:\backup\mydemo.db" -a "c:\backup\mydemo.log"
```

## 28.5.2 -ad Database Option

Specifies the directories containing transaction log files to be applied to the database.

### ⌘ Syntax

```
start_iq [ <server-options> ] <database-file> -ad <log-directory-specification> ...
```

## Applies to

All operating systems.

## Remarks

`<log-directory-specification>` can be one or more directory locations. When specifying more than one directory, use a semicolon as the delimiter.

When you include the `-ad` option, the specified directories are scanned for transaction log files associated with the database. Appropriate transaction log files with starting log offsets greater than or equal to the start log offset stored in the database file are applied, in log offset order. For point-in-time recovery, changes are applied starting from the time stamp or offset specified.

Once all the transaction log files have been applied, the database is stopped. Specify the `-as` database server option if you want the database to continue running once the transaction log files have been applied.

If any of the log directories are specified using a relative path, then this path is interpreted as relative to the `<database-file>` directory.

The `-ad` database option must be specified after the `<database-file>`, and applies only to that database.

## Example

The database server applies the transaction log files in the `c:\backup` directory to the `mysample.db` database and then stops the database once the transaction log files have been applied.

```
start_iq "c:\mysample.db" -ad "c:\backup"
```

The database server applies the transaction log files in the `c:\backup` and `c:\backup2` directories to the `mysample.db` database and the database continues running once the transaction log files have been applied.

```
start_iq "c:\mysample.db" -ad "c:\backup;c:\backup2" -as
```

## 28.5.3 -al Database Option

Extends `LOGIN_MODE` for `LDAPUA` only to a select number of users using Standard authentication.

### ↔ Syntax

```
start_iq -al <"user1;user2;user3" server_name.cfg database_name.db>
```

## Remarks

- Up to five user IDs can be specified, separated by semi-colons, and enclosed in double quotation marks.
- When run at the database level, it remains in effect until the next time the database is stopped/started.

## 28.5.4 -ar Database Option

Specifies that any transaction log files located in the same directory as the current transaction log should be applied to the database.

### ⌘ Syntax

```
start_iq [ <server-options> ] <database-file> -ar ...
```

### Applies to

All operating systems.

### Remarks

When you include the -ar option, the database server looks for transaction log files associated with the database that are located in the same directory as the current transaction log. The transaction log location is obtained from the database. Transaction log files with starting log offsets greater than or equal to the start log offset stored in the database are applied, in log offset order. Once all the transaction log files have been applied, the database is stopped. You must also specify the -as option if you want the database to continue running once the transaction log files have been applied.

The -ar database option must be specified after the <database-file>, and applies only to that database.

### Example

The database server applies the transaction log files (whose location is obtained from the database) to the `mysample.db` database. The database continues running after the transaction log files have been applied.

```
start_iq "c:\mysample.db" -ar -as
```

### Related Information

[-a Database Option \[page 210\]](#)

[-ad Database Option \[page 211\]](#)

[-as Database Option \[page 214\]](#)

## 28.5.5 -as Database Option

Specifies that the database should continue to run after transaction logs have been applied (used in conjunction with -ad or -ar).

### ⌘ Syntax

```
start_iq [ <server-options> ] <database-file> { -ad <log-dir> | -ar } -as ...
```

### Applies to

All operating systems.

### Remarks

The -as option must be specified in conjunction with either the -ad or -ar option. When you include -as, the database continues running after the transaction logs are applied to it.

The -as database option must be specified after the <database-file>, and applies only to that database.

### Example

The database server applies the transaction log files to the `mysample.db` database. In this case, because -ar is specified, the database server obtains the location of the transaction logs from the database. The database continues running after the transaction log files have been applied.

```
start_iq "c:\mysample.db" -ar -as
```

The database server applies the transaction log files in the `backup` directory to the `mysample.db` database. The database continues running after the transaction log files have been applied.

```
start_iq "c:\mysample.db" -ad "c:\backup" -as
```

### Related Information

[-a Database Option \[page 210\]](#)

[-ad Database Option \[page 211\]](#)

[-ar Database Option \[page 213\]](#)

## 28.5.6 -dh Database Option

Makes a database undetectable when the Server Location utility `dblocate -d` is run against the server.

### ↳ Syntax

```
start_iq -dh
```

## 28.5.7 -ds Database Option

Specifies the directory where the dbspaces for the database and the transaction log are located.

### ↳ Syntax

```
start_iq <database-file> -ds <dbspace-directory> ...
```

## Applies to

All operating systems.

This option only applies to Catalog dbspaces, not IQ dbspaces.

## Remarks

When a dbspace directory is specified, the database server only searches this directory for dbspaces. The location of the dbspace appears in the database server messages window.

If your backup includes dbspaces with full path names, you can use this option to start the backed up copy of the database on the same computer as the original database while the original database is still running.

The `-ds` database option must be specified after the `<database-file>`, and applies only to that database.

If a transaction log file is not found in the directory specified by this option, then one is created in this location.

### ⚠ Caution

The `-ds` option should only be used for recovery. If you specify this option and the database has a current, live transaction log that is not located in the directory specified by the `-ds` option, then a new transaction log is created in the specified location.

## Example

The following example starts a database server that looks for dbspaces in the directory `c:\backup\Nov15`:

```
start_iq c:\backup\Nov15\my.db -ds c:\backup\Nov15\
```

The following example starts a database server that looks for dbspaces in the current directory:

```
start_iq my.db -ds .
```

## 28.5.8 -ek Database Option

Specifies the database encryption key.

### ↳ Syntax

```
start_iq -ek <key>
```

## Remarks

Provided after the file name of a strongly encrypted database. Requires the key value as an argument to start an encrypted database. The key value is a string, including mixed cases, numbers, letters, and special characters. If you have a strongly encrypted database, you must provide the encryption key to use the database or transaction log; if you do not, the command fails. For a strongly encrypted database, you must specify either `-ek` or `-ep`, but not both.

## 28.5.9 -iqfreq Database Option

Marks the specified database as in use and restores the IQ main store portion of the database to its last known consistent state.

### ↳ Syntax

```
start_iq -iqfreq <dbname>
```

## Remarks

Do not use `-iqfrec` during normal operations; use it only while force-recovering a database after seeing `s_buf` or free list errors during recovery after SAP IQ server failure. The `<dbname>` must be the physical database name, not a logical name or nickname.

### Note

The option `-iqfrec` applies only to the IQ main store portion of the database, not to the IQ catalog store. `-iqfrec` does not enable a forced recovery on the catalog store.

Follow correct system recovery and database repair procedures when using `-iqfrec`.

## 28.5.10 -m Database Option

A server startup option that truncates (deletes) the transaction log when a checkpoint is done, either at shutdown or as a result of a checkpoint scheduled by the server.

### Syntax

```
start_iq -m
```

## Remarks

Truncation provides a way to automatically limit the growth of the transaction log. Checkpoint frequency is still controlled by the `CHECKPOINT_TIME` and `RECOVERY_TIME` options (also definable on the command line).

Avoid setting the `-m` option in a production environment, where transaction logs are required for recovery. Transaction log truncation is normally used in a situation where you have two configuration files:

- A configuration file for normal SAP IQ server startup.
- A separate configuration file, limited to the DBA user, for starting SAP IQ with the `-m` option set.

In this scenario, the DBA:

1. Backs up the database.
2. Shuts down the SAP IQ server.
3. Starts SAP IQ with the configuration file containing the `-m` option, which truncates the transaction log.
4. Shuts down the SAP IQ server.
5. Restarts the server with the normal configuration file.

To avoid database file fragmentation, place the transaction log on a separate device or partition than the database itself.

This option is the same as the `-m` server option, but applies only to the current database or the database identified by the `<database-file>` command line variable.

## 28.5.11 -n Database Option

Provides an alternate name, or nickname, for the database.

### ↔ Syntax

```
start_iq -n <name>
```

### Remarks

Using a nickname simplifies connections. For Open Client, the `-n` nickname must be the same as the entry in the `interfaces` file.

### ⓘ Note

Since a database server can load several databases, the database name is used to distinguish the different databases. However, run only one database on an SAP IQ server. If you must run two databases, start two SAP IQ database servers on different ports.

By default, the database receives as a name the file name with the path and extension removed. For example, you start a server on `/IQ/IQ-16_1/demo/iqdemo.db` and do not specify the `-n` option, then the name of the database is `iqdemo`. To avoid using the default name, always specify a server name.

For naming conventions, see the `-n` server option.

### ⓘ Note

There are two `-n` switches. If `-n` does not follow a database file name, the option names the server. If `-n` appears after a database file name, the switch is a database switch.

## 28.5.12 -r Database Option

Forces all databases that start on the database server to be read-only.

### ↔ Syntax

```
start_iq -r
```

### Remarks

No changes to the database are allowed: the database server doesn't modify the database file.

## 28.5.13 -sm Database Option

Provides an alternate database server name that can be used to access the read-only mirror database.

### ↳ Syntax

```
start_iq -sm <name>
```

### Remarks

The alternate-server-name is only active when the database server is acting as mirror for the database. By using the `-sm` and `-sn` command-line options, an application can always connect to the database on the primary or the mirror server, without knowing which physical server is acting as primary or mirror.

## 28.5.14 -sn Database Option

Provides an alternate server name for a single database running on a database server.

### ↳ Syntax

```
start_iq -sn <name>
```

### Remarks

The database server can be configured to listen for more than one server name for a particular database server. Server names other than the real server name are called alternate server names, and are specific to a particular database running on the database server. Clients using the alternate server name to connect can only connect to the database that specified the alternate server name. Alternate server names must be unique on the network; otherwise, the database fails to start. If the database is started in the server command and the alternate server name is not unique, the server fails to start.

## 28.5.15 -xp Database Option

Provides information to an operational server that allows it to connect to its partner and to the arbiter when database mirroring is being used.

### ↳ Syntax

```
start_iq [ <server-options> ] <database-file> -xp on
```

## Remarks

The `-xp` database option must be specified after the `<database-file>`, and applies only to that database.

If the connection parameters specified in the `-xp` option are invalid, and there are multiple databases running on the server, then the mirrored database fails to start and does not attempt to reconnect. If the mirrored database is the only database running on the database server, then the database server does not start.

## ON

You can only use database mirroring if you specify the `-xp` option when the database server is started, even if mirroring information is stored in the database. When you specify `-xp <on>`, you cannot specify other mirroring options with the `-xp` option. The value `off` is not supported. Database mirroring settings are defined in the database using the following statements:

- `CREATE MIRROR SERVER`
- `SET MIRROR OPTION`

When you specify `-xp <on>`, you should also specify the name of the database server in the mirroring system using the `-n` option. It is also recommended that you also include the `-su` option to specify the password for the utility database. Then, you can use the utility database to shut down the database server, or force the mirror server to become the primary server, if necessary.

# 29 stop\_iq Database Shutdown Utility

On UNIX and Linux platforms, you can stop the database server using the `stop_iq` utility.

Use `stop_iq` to stop your server and close all user connections to your server.

`stop_iq` shuts down a server without regard for user connections or load process status. Normally, do not shut down a server while it is still connected to one or more clients.

## Note

The `dbstop` utility offers a finer level of control, providing options to control whether a server is stopped based on user connections.

### In this section:

#### [When to Stop and Restart the Server \[page 221\]](#)

In some situations, the server may need to be stopped and restarted.

#### [Ways to Stop Database Servers \[page 222\]](#)

The preferred methods of stopping the database server in SAP IQ are as follows:

#### [Permissions Required to Stop the Server \[page 224\]](#)

When you start a server, you can use the `-gk` option to set the level of permissions required for users to stop the server with `DBSTOP` or `STOP ENGINE`.

#### [Stopping the Database Server with stop\\_iq \[page 224\]](#)

Run `stop_iq` from the command line.

## 29.1 When to Stop and Restart the Server

In some situations, the server may need to be stopped and restarted.

For example:

- To install a new version of SAP IQ
- To reset some server command line options
- To cause a small number of server-wide database options to take effect
- Before closing the operating system session

## 29.2 Ways to Stop Database Servers

The preferred methods of stopping the database server in SAP IQ are as follows:

- In UNIX, use the `stop_iq` utility at the operating system command line. When you run `stop_iq`, the following message displays:

```
"Please note that 'stop_iq' will shut down a server completely without regard for users, connections, or load process status. For more control, use the 'dbstop' utility, which has options that control stopping servers based on active connections."
```

- In Windows, click **Shutdown** on the database server display or right-click the IQ icon in the system tray and select **Exit**.
- In Windows, if the server is run as a service, open the Service Manager in Control Panel. Select the service and click **Stop**.

Avoid shutting down a server that is still connected to one or more clients. Doing so returns a warning that any uncommitted transactions will be lost. Disconnect or close all the clients and try again.

You can also stop the database server in the following ways:

- At the operating system command line, issue the `DBSTOP` command with appropriate parameters. Use the same parameters as when you started the server. Without the proper connection parameters `DBSTOP` does not know how to connect to the server to tell it to shut down.
- In an Interactive SQL window or command file, issue the `STOP ENGINE` command to stop a named database server.
- In UNIX, in the window where the database server was started, type:

```
q
```

This command does not work if you have redirected input to a different device or if you started the server with `start_iq`. It only works with `iqsrv16`.

- In a UNIX `cron` or `at` job, use `stop_iq` with the appropriate `-stop` option. The utility stops one or all servers associated with the user who starts the `cron` or `at` job depending on the parameter specified. The user must be the same one who started the server. No operator prompting occurs, and no operator action is required.

To use `stop_iq` in such jobs, specify the utility with the appropriate `-stop` option:

```
stop_iq -stop one
```

Setting `-stop one` shuts down a single server, when exactly one running server was started by the user ID that starts the `cron` or `at` job.

```
stop_iq -stop all
```

Setting `-stop all` shuts down all servers that were started by the user ID that starts the `cron` or `at` job.

### Note

You must specify the full path name to the `stop_iq` executable in the `cron` statement.

## Example — Stop a Server With stop\_iq

The following example uses the `stop_iq` utility in a UNIX operating system command line to shut down a SAP IQ server and close all user connections to it.

When you issue the `stop_iq` command, SAP IQ lists all the servers owned by other users, followed by the server(s) you own. It then asks if you want to stop your server. For example:

```
% stop_iq

Checking system for IQ 16 Servers ...
The following 2 server(s) are owned by other users.
##      Owner      PID      Started      CPU_Time      Additional Information
--      -
      handari 19895 15:43:44 183:38
start_iq @iqdemo.cfg iqdemo.db -gn 105 -o /server1/users/surya/IQ-16_1/logfiles/
surya_ibm2.001.srvlog -hn 8          pamela 409802 18:05:02 0:05
SVR:ibml_iqdemo2 DB:iqdemo
PORT:2678/ibml/users/IQ/iql6/IQ-16_1/bin64/iqsrv16 @iqdemo.cfg
iqdemo.db -ti 4400 -gn 25 -o /ibml/users/IQ/iql6/IQ-16_1/logfiles/ibm64qa_iq
The following 1 server(s) are owned by 'kermit'
##      Owner      PID      Started      CPU_Time      Additional Information
--      -
1:      kermit 422034 15:11:37 0:07 SVR:myserver_iqdemo
DB:iqdemo PORT:2638 /myserver/users/IQ/iql6/IQ-16_1/bin64/iqsrv16
@iqdemo.cfg iqdemo.db -ti 4400 -gn 25 -o /myserver/users/IQ/iql6/IQ-16_1/
logfiles/myserver_iq
start_iq -c 32m -gd all -gm 10 -gn 25 -gp 4096 -ti 4400 -tl 300 @iqdemo.cfg
--

Please note that 'stop_iq' will shut down a server completely
without regard for users connections or load processes status.
For more control, use the 'dbstop' utility, which has options
that control stopping servers based on active connections.
Do you want to stop the server displayed above <Y/N>?
```

To shut down the server, type `Y` (yes). Messages like the following display:

```
Shutting down server (422034) ...
Checkpointing server (422034) ...
Server shutdown.
```

To leave the server running, type `N` (no). You return to the system prompt and IQ does not shut down the server.

If no running servers were started by your user ID, SAP IQ displays information about servers run by other users, then a message like the following:

```
There are no servers owned by 'kermit'
```

## Example — Stop a Server From Interactive SQL

The following example stops a server from Interactive SQL:

```
STOP ENGINE Ottawa UNCONDITIONALLY
```

The optional keyword `UNCONDITIONALLY` specifies that the database server will be stopped even if there are connections to it.

### Note

You can stop a server from Interactive SQL if you are connected as DBA to one of the databases running on that server (including the `utility_db` database), or if the server was started with the `-gk ALL` option.

## 29.3 Permissions Required to Stop the Server

When you start a server, you can use the `-gk` option to set the level of permissions required for users to stop the server with `DBSTOP` or `STOP ENGINE`.

The default level of permissions required is `DBA`, which requires the `SERVER OPERATOR` system privilege, but you can also set the value to `ALL` or `NONE`. If you set it to `NONE`, even the a user with `SERVER OPERATOR` system privileges cannot execute `STOP ENGINE`. In a production environment, only the DBA should be allowed to stop the database server.

Running `stop_iq` at the UNIX command line, or Shutdown on Windows platforms, still allows you to stop the server and databases on the machine where the server was started.

## 29.4 Stopping the Database Server with `stop_iq`

Run `stop_iq` from the command line.

### Procedure

1. Issue a command in this format:

```
stop_iq [ -cleanup ] [ -stop [ one | all ] ] [ -user <user_name> ] [
-version [ 12 | 15 | all ]> ] [ -wait <seconds> ]
```

2. When prompted to stop your server, respond `Y` (yes).
3. If the server is connected to one or more clients, respond to the warning that uncommitted transactions will be lost.

### Results

The following example illustrates how to use `stop_iq` interactively to list all the servers that are running, and then shut down one server.

```
% stop_iq
```

```

Checking system ...
The following 1 server(s) are owned by 'TEST'
##      Owner      PID      Started  CPU_Time  Additional Information
--      -
1:     TEST       22399   08:56:39    1:43   SVR:QA_sun7qa DB:iqdemo PORT:8888
/sun7qa1/users/QA/090513/IQ-16_1/bin64/iqsrv16 @iqdemo.cfg iqdemo.db -ti 4400
--

Please note that 'stop_iq' will shutdown a server completely
without regard for users connections or load processes status.
For a finer level of detail the utility 'dbstop' has the options
to control whether a server is stopped based on active connections.
Do you want to stop the server displayed above <Y/N>? Y

```

```

Shutting down server (22399) ...
Checkpointing server (22399) ...
Server shutdown.

```

### In this section:

[stop\\_iq Options \[page 225\]](#)

This table lists the options available for the `stop_iq` utility.

[Stopping Servers in a cron or at Job \[page 226\]](#)

If using `stop_iq` in a cron or at job, use the appropriate `-stop` option:

[Servers with Long Paths \[page 227\]](#)

Depending on the operating system, the `stop_iq` utility may fail to report running servers when the path to the server exceeds 74 characters.

## 29.4.1 stop\_iq Options

This table lists the options available for the `stop_iq` utility.

Table 1: stop\_iq Options

Switch	Description
<code>-cleanup</code>	Removes the orphan SAP IQ process on Linux.
<code>-stop [ one   all   pid &lt;PID&gt;   server &lt;servername&gt; ]</code>	<p>Removes user interaction with <code>stop_iq</code>. Assumes a yes response to the prompts returned when you run <code>stop iq</code> interactively. See <i>Stopping the Database Server with stop_iq</i>.</p> <ul style="list-style-type: none"> <li>• <code>all</code> - stops all running servers.</li> <li>• <code>one</code> - stops one server (if only one, owned by you, is running).</li> <li>• <code>pid</code> - stops the server running on the UNIX process specified.</li> <li>• <code>server</code> - stops the named server.</li> </ul> <p>Server name can be up to 4294967295 bytes.</p>

Switch	Description
<code>-user</code>	<p>Performs two functions:</p> <ul style="list-style-type: none"> <li>If, due to truncation or substitution, <code>stop_iq</code> cannot find the server or agent owned by the current user, the system manager can specify the name and id found in the process table in the <code>-user</code> argument to shut down the server/agent with the <code>stop_iq</code> utility.</li> <li>A user with root privileges can shut down another user's server or agent without having to log in as that user. The <code>stop_iq</code> utility has no superuser (su) or root powers, so a nonprivileged user cannot shut down a server owned by another user.</li> </ul>
<code>-version</code>	Specifies the major version of SAP IQ that is being used. The default is the current major version (16.1).
<code>-wait</code>	Specify the time to wait for the server to shut down before timeout expires.

## 29.4.2 Stopping Servers in a cron or at Job

If using `stop_iq` in a cron or at job, use the appropriate `-stop` option:

### **-stop one**

Shuts down a single server, when exactly one running server was started by the user ID that starts the cron or at job. This prevents the wrong server from being shut down if several are running.

### **-stop all**

Shuts down all servers that were started by the user ID that starts the server.

For example:

```
stop_iq -stop one
```

```
stop_iq -stop all
```

### **Note**

In a cron statement, you must specify the complete path name to the `stop_iq` executable.

## 29.4.3 Servers with Long Paths

Depending on the operating system, the `stop_iq` utility may fail to report running servers when the path to the server exceeds 74 characters.

As a workaround, use a `ps -ef` command to display servers with long paths. For example:

```
ps -ef|grep myserver
rsmithson 1133      1    0 07:04:32 ?                223:35
/sunsys1234/users/rsmithson/mybigtest1234_withdeletion
_allcol/IQ-16_1/bin64/myserver
rsmithson 2046    862    0 10:02:30 pts/3          0:00 grep
myserver
```

# 30 Appendix: dbisqlc Interactive SQL Classic Utility (Deprecated)

The Interactive SQL Classic (`dbisqlc`) utility executes SQL statements against a database. This utility is deprecated.

The utility is similar to the Interactive SQL utility, except that it is not implemented in Java. The absence of Java can be an advantage if you are deploying the utility to a computer with limited resources.

## Note

Interactive SQL Classic is deprecated; however, there are currently no plans to remove it. Interactive SQL Classic is provided for backwards compatibility for running SQL scripts and as a lightweight tool for deployment. It does not support all the features that Interactive SQL supports and may not support all the features available in the current version of SAP IQ. Use the Interactive SQL utility instead.

The Interactive SQL Classic utility is supported on Microsoft Windows and UNIX.

### In this section:

[Interactive SQL Classic Syntax \[page 228\]](#)

Invoke Interactive SQL Classic from a command prompt.

[Function and Special Keys \(UNIX\) \[page 229\]](#)

Use function keys and special keys to move data and list database tables.

[Function and Special Keys \(Windows\) \[page 230\]](#)

Use function keys and special keys to move data and list database tables.

[Command Recall Keys \(Windows\) \[page 231\]](#)

Use key sequences to recall previous commands.

## 30.1 Interactive SQL Classic Syntax

Invoke Interactive SQL Classic from a command prompt.

```
dbisqlc [ <options> ] [ <dbisqlc-command > | <command-file> ]
```

If you specify `<dbisqlc-command>`, Interactive SQL Classic executes the command. You can also specify a command file name. If you do not specify `<dbisqlc-command>` or `<command-file >` argument, Interactive SQL Classic enters interactive mode, where you can type a command into a command window.

### In this section:

[Interactive SQL Classic Options \[page 229\]](#)

Specify these options when invoking Interactive SQL Classic from a command prompt.

## 30.1.1 Interactive SQL Classic Options

Specify these options when invoking Interactive SQL Classic from a command prompt.

Interactive SQL Classic option	Description
<code>-c &lt;"keyword=value; ..."&gt;</code>	Specifies connection parameters. If you do not specify any connection parameters, the environment variable SQLCONNECT is used. If Interactive SQL cannot connect, enter the appropriate parameters in the dialog box that appears.
	<b>Note</b> Always specify connection parameters for Interactive SQL Classic instead of relying on defaults, whether you specify them in a command line or an initialization file such as <code>.odbc.ini</code> on UNIX, or <code>odbc.ini</code> on Windows. If you start more than one database on a server, for example, specify the database name, and in a network with subnets, specify the communications protocol parameter with host number.
<code>-d &lt;delimiter&gt;</code>	Specifies a command delimiter. By default, the delimiter is the semi-colon.
<code>-q</code>	Quiet mode—does not display output messages. This option is useful only if you start Interactive SQL Classic with a command or command file.
<code>-r</code>	Returns the error "Not enough fields allocated in sqlda" if the defined result set of the stored procedure does not match the actual result set. This option may be useful when you are querying stored procedures.
<code>-x</code>	Checks syntax only. Scans commands but does not execute them. You may find this option useful for checking long command files for syntax errors.

## 30.2 Function and Special Keys (UNIX)

Use function keys and special keys to move data and list database tables.

Interactive SQL Classic function key (or special key) (UNIX)	Description
F5	Move data to the left by one column in the data window.
Shift+F5	Move data to the left by one character.
F6	Move data to the right by one column.
Shift+F6	Move data to the right by one character.

**Interactive SQL Classic function  
key (or special key) (UNIX)**

**Description**

F7	Display a list of the tables in the database. Use the up and down arrow keys to scroll through the table names changing the highlighted table name. While the list appears, press Enter to insert the current table name into the command window at the cursor position. While the list appears, press F7 to see a list of columns for the highlighted table. Again, use Enter to select the highlighted column name and put it into the command window at the cursor position.
Ctrl+PgUp	Move to the top of data.
Ctrl+PgDn	Move to bottom of data.

## 30.3 Function and Special Keys (Windows)

Use function keys and special keys to move data and list database tables.

**Interactive SQL Classic  
function key (or special key)  
(Windows)**

**Description**

F5	Move data to the left by one column in the data window.
Shift+F5	Move data to the left by one character.
F6	Move data to the right by one column.
Shift+F6	Move data to the right by one character.
F7	Display a list of the tables in the database. Use the up and down arrow keys to scroll through the table names changing the highlighted table name. While the list appears, press Enter to insert the current table name into the command window at the cursor position. While the list appears, press F7 to see a list of columns for the highlighted table. Again, use Enter to select the highlighted column name and put it into the command window at the cursor position.
F9	Execute the command that is in the command window. This operation can also be performed with the mouse by clicking Execute.
F10	Activate the menus at the top of the window.
Page Up	Move data up a page.
Page Down	Move data down a page.
Ctrl+PageUp	Move to top of data.

Interactive SQL Classic  
function key (or special key)  
(Windows)

Description

---

Ctrl+PageDown	Move to bottom of data.
---------------	-------------------------

---

## 30.4 Command Recall Keys (Windows)

Use key sequences to recall previous commands.

Interactive SQL Classic recall  
keys

Description

---

Ctrl+r	Brings up the command recall window
--------	-------------------------------------

---

Ctrl+p	Cycles backwards through previously executed commands. Retrieved commands are placed into the command window
--------	--

---

Ctrl+n	Cycles forward through previously executed commands
--------	---



---

# Important Disclaimers and Legal Information

## Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

## Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

## Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

## Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

## Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.



© 2026 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.

