SQL Anywhere® 16
Changes and Upgrading

Version 16.0
May 2014
Contents

About this book ........................................................................................................ vii

What's new in version 16.0 ...................................................................................... 1
  SQL Anywhere 16.0 Support Package enhancements ........................................ 1
  Product-wide new features ............................................................................... 7
  Product-wide behavior changes ....................................................................... 12
  Product-wide deprecated and removed features .............................................. 13
  SQL Anywhere new features ........................................................................... 14
  SQL Anywhere behavior changes .................................................................... 32
  SQL Anywhere deprecated and discontinued features ..................................... 39
  MobiLink new features .................................................................................... 42
  MobiLink behavior changes ........................................................................... 50
  QAnywhere new features ................................................................................ 54
  Relay Server new features ............................................................................. 54
  Relay Server behavior changes ....................................................................... 55
  UltraLite new features ..................................................................................... 57
  UltraLite behavior changes and deprecated features ..................................... 65
  Administration tools new features .................................................................. 66

What's new in version 12.0.1 ................................................................................. 73
  12.0.1 addendum: New features and behavior changes .................................. 73
  SQL Anywhere new features ........................................................................... 83
  Performance enhancements ............................................................................. 88
  Properties and Performance Monitor statistics .............................................. 89
  SQL Anywhere behavior changes ................................................................... 91
  SQL Anywhere deprecated and discontinued features ................................... 93
  MobiLink new features ................................................................................... 94
  MobiLink behavior changes .......................................................................... 98
  SQL Remote new features ............................................................................ 101
  UltraLite new features .................................................................................. 101
  UltraLite behavior changes and deprecated features .................................. 106
<table>
<thead>
<tr>
<th>Feature</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration tools new features</td>
<td>107</td>
</tr>
<tr>
<td>Documentation enhancements</td>
<td>112</td>
</tr>
<tr>
<td>What's new in version 12.0.0</td>
<td>113</td>
</tr>
<tr>
<td>Product-wide new features</td>
<td>113</td>
</tr>
<tr>
<td>Product-wide behavior changes</td>
<td>113</td>
</tr>
<tr>
<td>SQL Anywhere new features</td>
<td>114</td>
</tr>
<tr>
<td>SQL Anywhere behavior changes</td>
<td>140</td>
</tr>
<tr>
<td>SQL Anywhere deprecated and discontinued features</td>
<td>146</td>
</tr>
<tr>
<td>MobiLink new features</td>
<td>148</td>
</tr>
<tr>
<td>MobiLink behavior changes</td>
<td>157</td>
</tr>
<tr>
<td>SQL Remote new features</td>
<td>162</td>
</tr>
<tr>
<td>UltraLite new features</td>
<td>162</td>
</tr>
<tr>
<td>UltraLite behavior changes</td>
<td>167</td>
</tr>
<tr>
<td>UltraLiteJ new features</td>
<td>169</td>
</tr>
<tr>
<td>UltraLiteJ behavior changes and deprecated features</td>
<td>174</td>
</tr>
<tr>
<td>Administration tools new features</td>
<td>175</td>
</tr>
<tr>
<td>Documentation enhancements</td>
<td>185</td>
</tr>
<tr>
<td>What's new in version 11.0.1</td>
<td>187</td>
</tr>
<tr>
<td>SQL Anywhere new features</td>
<td>187</td>
</tr>
<tr>
<td>SQL Anywhere behavior changes and deprecated features</td>
<td>193</td>
</tr>
<tr>
<td>MobiLink new features</td>
<td>197</td>
</tr>
<tr>
<td>MobiLink behavior changes and deprecated features</td>
<td>198</td>
</tr>
<tr>
<td>UltraLite new features</td>
<td>199</td>
</tr>
<tr>
<td>UltraLite behavior changes and deprecated features</td>
<td>199</td>
</tr>
<tr>
<td>UltraLiteJ new features</td>
<td>199</td>
</tr>
<tr>
<td>UltraLiteJ behavior changes and deprecated features</td>
<td>200</td>
</tr>
<tr>
<td>Administration tool new features</td>
<td>200</td>
</tr>
<tr>
<td>Administration tool behavior changes and deprecated features</td>
<td>202</td>
</tr>
<tr>
<td>Product-wide new features</td>
<td>202</td>
</tr>
<tr>
<td>Documentation enhancements</td>
<td>203</td>
</tr>
<tr>
<td>What's new in version 11.0.0</td>
<td>205</td>
</tr>
</tbody>
</table>
About this book

This book describes new features in SQL Anywhere 16 and in previous versions of the software.

For information about new features and behavior changes in SQL Anywhere versions 10.0.1 and earlier, go to http://dcx.sybase.com/html/dbwnen10/dbwnen10.html.
What's new in version 16.0

For information about new features and behavior changes in versions of SQL Anywhere before version 10, see http://dcx.sybase.com/html/dbwnen10/dbwnen10.html.

SQL Anywhere 16.0 Support Package enhancements

[ This topic has been updated for build 1823. ]

The following enhancements have been made to SQL Anywhere 16.0 since its original release.

SQL Anywhere enhancements and behavior changes

- **xp_getenv system procedure improvement—database upgrade or rebuild required**
  The xp_getenv system procedure has been changed so that it is a procedure that runs with invoker privileges regardless of the invoker/definer setting. The procedure returns a LONG NVARCHAR value. Previously, it returned a LONG BINARY value.

  See “xp_getenv system procedure” [SQL Anywhere Server - SQL Reference].

- **sa_cpu_topology system procedure improvement—database upgrade or rebuild required**
  The sa_cpu_topology system procedure has been modified to include information about user-selected physical processors specified using the -gta database server option or the ProcessorAffinity server property. This feature allows the database server to make use of newly added processors during runtime (also know as hot-add and hot-remove).

  The restrictions and limitations for the -gt, -gta, and -gtc database server options are preserved.

  See “sa_cpu_topology system procedure” [SQL Anywhere Server - SQL Reference].

- **sp_list_directory system procedure enhancements**
  In addition to returning the path, file type, and file name of all files and directories in a specified location, the sp_list_directory system procedure now also returns the date the file was created, last modified, and last accessed, as well as the owner and any access permissions required for the file or directory.

  See “sp_list_directory system procedure” [SQL Anywhere Server - SQL Reference].

- **Multiple OData Producer support**

  [ This feature has been updated for build 1823. ]

  The OData Server now supports multiple customizable OData Producers that allow you to establish multiple database connections. Use the new embedded HTTP server option, Producers, in your OData Server configuration to create OData Producers.

  See “How to configure the OData Server” [SQL Anywhere Server - Programming].
Optimistic Concurrency Control (ETags)

OData Producers now support Optimistic Concurrency Control as defined by version 2.0 of the OData Specification. The concurrencytoken clause of the ENTITY OSDL statement is used to generate ETags that identify the state of an entity instance at the time the instance is requested.

See “ENTITY statement” [SQL Anywhere Server - Programming].

OData enhancement: Search strings are restricted to 254 bytes

When using OData filters—such as startswith, substringof, and indexofwith—on long search strings, searches are performed on the first 254 bytes only.

OData enhancement: Support for proxy table insertions

All key properties must be explicitly specified when creating entities in entity sets that are proxy tables in a SQL Anywhere database.

OData enhancement: substringof return value

The substringof(s1, s2) filter returns whether the s1 string is a substring of s2.

New database properties

The following database properties have been added in this release:

- BackupInProgress
- TimeWithoutClientConnection

See “List of database properties” [SQL Anywhere Server - Database Administration].

Support for callbacks in the SQL Anywhere C API

Support for callbacks has been added to version 3 of the SQL Anywhere C API. The following function is now available when _SACAPI_VERSION is defined as 3.

\[
\text{sacapi_bool sqlany_register_callback(a_sqlany_connection * sqlany_conn, a_sqlany_callback_type index, SQLANY_CALLBACK_PARM callback)};
\]

This function can be used to register callback functions.

JDBC

The SQL Anywhere JDBC driver support for escape sequences has been enhanced to include the TIMESTAMPADD and TIMESTAMPDIFF functions.

See “ODBC escape syntax” [SQL Anywhere Server - Programming].

ODBC

The ODBC driver support for escape sequences has been enhanced to include the TIMESTAMPADD and TIMESTAMPDIFF functions. Previously, calling the ODBC SQLGetInfo function to retrieve the version of the ODBC driver (SQL_DRIVER_VER) returned a string that did not include the build number of the driver. Now, the format of the string returned is xx.yy.zzzz where xx is the 2-digit major version number, yy is the 2-digit minor version number, and zzzz is the build number (for example, 16.00.1234).
SQL Anywhere OLE DB provider enhancements—database upgrade required

Updates have been made to the SQL Anywhere OLE DB provider.

MobiLink enhancements and behavior changes

- **Support for SAP Sybase IQ 16** The MobiLink server now supports consolidated databases running on SAP Sybase IQ 16.0 servers. For information about recommended ODBC drivers, see http://www.sybase.com/detail?id=1011880.

- **dbmlsync offline transaction log retrieval has changed** The dbmlsync utility can now retrieve offline transaction logs from the SQL Anywhere database server instead of accessing them directly. If offline transaction logs are required but the given offline transaction log directory cannot be opened or it does not contain offline transaction log files, then dbmlsync retrieves the offline transaction logs through the database server. The following restrictions apply:

  ○ The user ID that is used by dbmlsync to connect to the synchronization database must have the READ FILE and WRITE FILE privileges and all the offline transaction log files must be in the online transaction log directory.

  ○ The SQL Anywhere database server must have Support Package build number 1823 or later to support this feature.

  There is a slight performance penalty when using this feature because the database server must do more work to retrieve the pages. If performance is critical, then using the dbmlsync OfflineDirectory extended option may be best for your deployment.

  See “OfflineDirectory (dir) extended option” [MobiLink - Client Administration].

- **dbmlsync provides the ability to restart downloads when no bytes of data have been received** The dbmlsync utility now allows you to restart a failed download even if no bytes of the download have been received. Previously a download could only be restarted if at least one byte had been received.

- **Synchronization models can be duplicated** The Synchronization Models menu in the MobiLink plug-in has a new option to duplicate a synchronization model.

Relay Server enhancements and behavior changes

- **Relay Server Record affinity flag now accepts a value of expired** The Relay Server Record affinity flag has been extended to allow Relay Server to signal that the affinity cookie is expired.

  See “Relay Server Record” [Relay Server].

Product-wide deprecated and removed features

- **Strong encryption in FIPS, TLS, and HTTPS now achieved using OpenSSL**

  These changes are being released in version 12 and version 16 Support Packages.

SQL Anywhere version 16.0 up to build 1695 and version 12.0.1 up to build 3986 included a Certicom encryption module that provided strong encryption used throughout the software. Now, SQL
Anywhere includes an OpenSSL encryption module for the strong encryption. The Certicom encryption module has been removed.

Read the following descriptions to determine how you may be impacted by this change.

○ **FIPS encryption now requires the private key of an identity file to be encrypted using AES**
  
  • OpenSSL FIPS supports AES encryption for the private key of an identity file. New servers using the OpenSSL FIPS encryption module will not start when using an identity file that has its private key encrypted with 3DES. You must re-encrypt the identity file using AES. To do this, run a command similar to the following using an upgraded viewcert utility:

  ```
  viewcert -p -o new-file-name -op new-password -ip old-password old-file-name
  ```

  The new and old passwords can be the same.

  • The sample server certificate (`rsaserver.id`) and client certificate (`rsaclient.id`) have been modified so that the private key is encrypted using AES rather than 3DES.

  • Versions of the server that use the Certicom encryption module will not start when using an identity file that has its private key encrypted using AES. Trusted root certificate files specified using `trusted_certificates` do not need to be modified.

○ **Self-signed certificates must now have the Certificate Signing attribute set**  
  Self-signed certificates must now have the Certificate Signing attribute set when using the identity encryption option (for example, the `-x mlsrvXX` and `-xs dbsrvXX` options). To determine if a certificate has the Certificate Signing attribute set, use the viewcert utility and look for Certificate Signing in the Key Usage portion of the output. If your self-signed certificates do not have the Certificate Signing attribute set, then you must regenerate the certificates.

○ **Create Certificate utility (createcert) now uses AES encryption instead of 3DES**  
  The Create Certificate utility (createcert) now uses AES rather than 3DES encryption for encrypting the private key in the server identity file.

  A new option, `-3des`, has been added to the Create Certificate utility. Use this option when you want to create a 3DES-encrypted server identity file that can be used by both new and old servers. Note that new servers running in FIPS mode cannot start using 3DES-encrypted certificates; however, if you are not running in FIPS mode, then you can use 3DES-encrypted certificates.

○ **View Certificate utility (viewcert) now uses AES encryption instead of 3DES**  
  The View Certificate utility (viewcert) now uses AES rather than 3DES encryption when you specify the `-p` option to PEM-encode the output and when you specify the `-ip` and `-op` options to set the password.

  A new option, `-3des`, has been added to the View Certificate utility to allow you encrypt output and passwords using 3DES instead of AES.

○ **Database server now loads the FIPS driver file, `dbfipsXX.dll`, at startup**  
  Previously, the 32-bit Windows database server loaded the FIPS driver file, `dbfipsXX.dll`, only when needed.
Now, the 32-bit Windows database server always attempts to load `dbfipsXX.dll` at startup, and keeps it loaded for the life of the server. If loading `dbfips16.dll` fails, then an error is returned only when an attempt is made to use FIPS encryption.

- **Deploying FIPS** If you are deploying FIPS encryption, then there are new shared libraries to deploy; these files are included in your software. The former files, `sbgse2.dll` and `libsbgse2.so`, are no longer installed by the software. The new files to deploy are:
  - Windows 64-bit: `libeay32.dll`, `ssleay32.dll`, and `msvcr100.dll`
  - Windows 32-bit: `libeay32.dll`, `ssleay32.dll`, and `msvcr90.dll`
  - Linux: `libcrypto.so` and `libssl.so`

**Note:**
On Windows, although 32-bit and 64-bit FIPS-certified OpenSSL libraries for encryption are provided, you must use the 64-bit libraries on a 64-bit system.

- **MobiLink-related changes and information**
  - **Connecting to a MobiLink server using client-side certificates now requires the Digital Signature certificate attribute to be set**
    TLS/SSL connections to a MobiLink server using client-side certificates now require the client-side certificate to have the Digital Signature attribute set. If the attribute is not set, then the connection will fail.

    To determine if a certificate has the Digital Signature attribute set, use the View Certificate utility (viewcert) and look for the Digital Signature attribute in the Key Usage portion of the output. If your client-side certificates do not have the Digital Signature attribute set, then you must regenerate the certificates.

  - **FIPS-based end-to-end encryption now requires the private key to be encrypted using AES**
    If the private key file provided to a MobiLink server by the `e2ee_private_key` file option of the `-x` command-line option is encoded using 3DES and you are running in FIPS mode, then the private key file needs to be regenerated with the private key encrypted using AES.

  - **How to update a MobiLink deployment that uses non-FIPS TLS/SSL (includes HTTPS) and client-side certificates**
    1. If your client-side identity certificates do not have the Digital Signature attribute set and the client connects directly to the MobiLink server, then you must regenerate and deploy client-side certificates with the Digital Signature attribute set.
    2. Update the server-side binaries.
    3. Update the client-side binaries.

  - **How to update a MobiLink deployment that uses FIPS, TLS/SSL (includes HTTPS) and client-side certificates**
    These steps update the client identity certificates twice if the Digital Signature attribute is missing from client-side identity certificates. This procedure can
make the update less disruptive because synchronizations can continue without having to coordinate the client-side and server-side updates to occur at the same time.

1. If your current client-side identity certificates do not have the Digital Signature attribute set and the client connects directly to the MobiLink server, then you must regenerate and deploy client-side certificates with the Digital Signature attribute set.

2. Update the server-side binaries (remembering to include the new FIPS driver files) and deploy server identity certificates with AES-encrypted private keys.

3. Update the client-side binaries (remembering to include the new FIPS driver files) and deploy client identity certificates with AES-encrypted private keys.

- How to update a MobiLink deployment that uses FIPS and end-to-end encryption

1. Regenerate the primary key file referenced by the e2ee_private_key encryption option.

2. Shut down the MobiLink server.

3. Update the MobiLink server binaries, remembering to include the new required FIPS driver files.

4. Change the e2ee_private_key option to point to the new private key file (or replace the old file), updating the e2ee_private_key_password, if required.

5. Restart the MobiLink server.

○ UltraLite UltraLite no longer supports deploying FIPS on Windows Mobile.

- FIPS-certified encryption on Windows Mobile no longer supported. Previously, FIPS-certified encryption was supported on Windows Mobile devices, but only devices that used ARM processors. This support has ended. FIPS-certified encryption is no longer supported for Windows Mobile.

- SAP JRE replaces Oracle JRE in the software Previously, an Oracle JRE was shipped with the software for use by clients. Now, an SAP JRE is shipped instead.

Upgrading overwrites the JRE directory (%SQLANY16%/binXX\jre170) and its subdirectories. If you are using certificates, then your certificate store (%SQLANY16%/binXX\jre170\lib\security\cacerts) is overwritten, including your certificates. Similarly, fonts you added to the %SQLANY16%/binXX\jre170\lib\fonts\fallback directory to help display characters in the administration tools may be lost. To minimize upgrading steps as a result of the JRE change, create a backup copy of the JRE directory and all of its subdirectories before you upgrade so that you can refer to or restore files (such as cacerts) from the backup, as needed.

The SAP JRE may perform differently than the Oracle JRE. Use the java_vm_options option (SQL Anywhere), and/or the -sl java option (MobiLink) to optimize your Java VM startup settings.
Product-wide new features

Following is a list of product-wide additions introduced in version 16.0. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

New security model: Role-based access control (RBAC)

SQL Anywhere 16.0 introduces a new role- and privileged-based security model to replace the former authorities and permissions security model. The new role-based security model provides you with granular control over the privileged tasks that users can perform, and simpler administration of access control.

For existing SQL Anywhere customers who are upgrading to version 16.0, backwards compatibility has been provided so that your applications will continue to function once your database has been upgraded. A special chapter has been provided for you to understand the difference between the models, what takes place automatically when you upgrade, and what you should consider doing if you have applications that use SQL Anywhere (such as updating your GRANT and REVOKE statement calls). See “Upgrading to role-based security” [SQL Anywhere Server - Database Administration].

Tutorials are available here: “Tutorial: Granting roles and privileges (Sybase Central)” [SQL Anywhere Server - Database Administration], and “Tutorial: Granting roles and privileges (SQL)” [SQL Anywhere Server - Database Administration].

Here is a brief overview of new and changed features provided in support of the new role-based security model.

- **Overview of role-based security model**  
  A complete set of system privileges and roles have been added to increase security by providing you precision over the capabilities you want your users to have. For every privileged operation that can be performed in the system, a system privilege has been created. Roles combine privileges into logical groups. SQL Anywhere provides you with many predefined roles, but you can also create your own roles. See “User security (roles and privileges)” [SQL Anywhere Server - Database Administration].

  If you are upgrading, authorities, permissions, and groups have been replaced with roles, privileges, and user-extended roles. See “Upgrading to role-based security” [SQL Anywhere Server - Database Administration].

- **New TRUNCATE object-level privilege**  
  A new object-level privilege, TRUNCATE, has been added to allow a user to truncate a specified table or materialized view. This privilege did not exist as an object-level permission in previous releases of the software. See “Object-level privileges” [SQL Anywhere Server - Database Administration].

- **New LOAD object-level privilege**  
  A new object-level privilege, LOAD, has been added to allow a user to load a specific table. This privilege did not exist as an object-level permission in previous releases of the software. See “Object-level privileges” [SQL Anywhere Server - Database Administration].

- **Automatic unlocking of user accounts**  
  The root_auto_lock_time (root login policy only) and auto_unlock_time login policy options allow you to determine the automatic unlocking time period for users locked out of the database due to failed login attempts. These options are available in the...
CREATE LOGIN POLICY and ALTER LOGIN POLICY statements. See “Automatic unlocking of user accounts” [SQL Anywhere Server - Database Administration].

● System procedures Following are the system procedures added in support of role-based security:

○ **sp_objectpermission system procedure** This procedure generates a report on the object privileges granted to the specified object, dspace, role, or user name. You must rebuild existing databases to get this system procedure. See “sp_objectpermission system procedure” [SQL Anywhere Server - SQL Reference].

○ **sp_displayroles system procedure** Returns all roles granted to the specified system privilege, system role, user-defined role, or user name, or displays the entire hierarchy tree of roles. See “sp_displayroles system procedure” [SQL Anywhere Server - SQL Reference].

○ **sp_has_role system procedure** Returns whether the invoker of the procedure has been granted the specified system privilege or user-defined role. See “sp_has_role system procedure” [SQL Anywhere Server - SQL Reference].

○ **sp_proc_priv system procedure** Returns the list of system privileges required to run a procedure. See “sp_proc_priv system procedure” [SQL Anywhere Server - SQL Reference].

○ **sp_auth_sys_role_info system procedure** Returns the mapping of authorities from previous versions of SQL Anywhere to their corresponding compatibility roles. See “sp_auth_sys_role_info system procedure” [SQL Anywhere Server - SQL Reference].

○ **sp_sys_priv_role_info system procedure** Returns the mapping of system privileges to their underlying system roles. See “sp_sys_priv_role_info system procedure” [SQL Anywhere Server - SQL Reference].

● Database options Following are the database options added in support of role-based security:

○ **min_role_admins option** Sets the minimum number of administrators required for a role. See “min_role_admins option” [SQL Anywhere Server - Database Administration].

○ **db_publisher option** This option stores the user ID of the database publisher. It can be set in the same way as other database options, but can also be set using the GRANT PUBLISH and REVOKE PUBLISH statements. See “db_publisher option” [SQL Anywhere Server - Database Administration].

● Database server options The behavior of the following database server options has been changed to support role-based security:

○ **-gu database server option** Setting the value to DBA means that only a user with the SERVER OPERATOR privilege can create or drop databases. See “-gu database server option” [SQL Anywhere Server - Database Administration].

○ **-gk database server option** Setting the value to DBA means that only a user with the SERVER OPERATOR privilege can shut down a database server with the dbstop utility. See “-gk database server option” [SQL Anywhere Server - Database Administration].
- **gd database server option**  Setting the value to DBA means that only a user with the SERVER OPERATOR privilege can start databases. See “-gd database server option” [SQL Anywhere Server - Database Administration].

- **gl database server option**  Setting the value to DBA means that only a user with the LOAD ANY TABLE or ALTER ANY TABLE privileges can execute the LOAD statement. The user must have the SELECT ANY TABLE privilege to execute the UNLOAD statement. See “-gl database server option” [SQL Anywhere Server - Database Administration].

- **New and changed SQL statements**  Following are the SQL statements added or changed in support of role-based security:

  - **Changes to the SELECT statement**  You can now specify a FOR JSON clause in a SELECT statement. The FOR JSON clause specifies that the result set is to be returned in JSON format. You can specify one of the following JSON modes:

    - **RAW**  Allows you to return each row in the query result set as a flattened JSON representation.

    - **AUTO**  Allows you to return the query result set as nested JSON objects based on query joins.

    - **EXPLICIT**  Allows you to specify columns as simple values, objects, and nested hierarchical objects to produce uniform or heterogeneous arrays.

    See “SELECT statement” [SQL Anywhere Server - SQL Reference] and “Use of the FOR JSON clause to retrieve query results as JSON” [SQL Anywhere Server - SQL Usage].

  - **Changes to the GRANT statement**  The GRANT statement has been enhanced to allow granting of roles and privileges. See “GRANT statement” [SQL Anywhere Server - SQL Reference].

    The old syntax for granting authorities, permissions, and membership in groups is still supported but deprecated. See “GRANT statement (authorities and groups) (deprecated)” [SQL Anywhere Server - Database Administration].

  - **Changes to the REVOKE statement**  The REVOKE statement has been enhanced to revoke roles and privileges. See “REVOKE statement” [SQL Anywhere Server - SQL Reference].

    The old syntax for granting authorities, permissions, and membership in groups is still supported but deprecated. See “REVOKE statement (authorities and groups) (deprecated)” [SQL Anywhere Server - Database Administration].

  - **New CREATE ROLE statement**  Creates or replaces a role, converts a user to a role, or manages role administrators on a role. See “CREATE ROLE statement” [SQL Anywhere Server - SQL Reference].

  - **New ALTER ROLE statement**  Migrates a compatibility role (roles that begin with SYS_AUTHOR) to a user-defined role, then drops the compatibility role. See “ALTER ROLE statement” [SQL Anywhere Server - SQL Reference].
- **New DROP ROLE statement**  Removes a user-defined or a compatibility role from the database, or converts a user-extended role to a regular user. See “DROP ROLE statement” [SQL Anywhere Server - SQL Reference]..

- **New GRANT ROLE SYS_RUN_REPLICATION_ROLE statement**  This statement grants the role required to run replication. See “GRANT ROLE SYS_RUN_REPLICATION_ROLE statement [MobiLink] [SQL Remote]” [SQL Anywhere Server - SQL Reference].

- **New GRANT ROLE SYS_REPLICATION_ADMIN_ROLE statement**  This statement grants the role required to administer replication. See “GRANT ROLE SYS_REPLICATION_ADMIN_ROLE statement [MobiLink] [SQL Remote]” [SQL Anywhere Server - SQL Reference].

- **Changes to GRANT PUBLISH and REVOKE PUBLISH statements**  Previously, the GRANT PUBLISH and REVOKE PUBLISH statements updated the database publisher information in the ISYSAUTHORITY system table. However, as part of role-based security, the database publisher user ID is now stored as the db_publisher database option. These statements now update the value of the db_publisher database option, instead of updating ISYSUSERAUTHORITY. See “GRANT PUBLISH statement [SQL Remote]” [SQL Anywhere Server - SQL Reference] and “REVOKE PUBLISH statement [SQL Remote]” [SQL Anywhere Server - SQL Reference].

- **Changes to the catalog**  Following are the catalog changes made in support of role-based security. The changes are made to the catalog table, but since you can only access the corresponding system view, the view is mentioned as well.

<table>
<thead>
<tr>
<th>Catalog item</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISYSROLEGRANT system table / “SYSROLE-GRANT system view” [SQL Anywhere Server - SQL Reference]</td>
<td>New. Stores information about role membership and type of membership</td>
</tr>
<tr>
<td>“SYSROLEGRANTS consolidated view” [SQL Anywhere Server - SQL Reference]</td>
<td>New. Same as SYSROLEGRANT, but includes two additional columns: role_name, and grantee_name.</td>
</tr>
<tr>
<td>“SYSGROUP compatibility view” [SQL Anywhere Server - SQL Reference]</td>
<td>This view was previously a system view, but is now a compatibility view.</td>
</tr>
<tr>
<td>“SYSGROUPS compatibility view” [SQL Anywhere Server - SQL Reference]</td>
<td>This view was previously a consolidated view, but is now a compatibility view.</td>
</tr>
<tr>
<td>“SYSUSER system view” [SQL Anywhere Server - SQL Reference]</td>
<td>New column: dual_password</td>
</tr>
<tr>
<td>Catalog item</td>
<td>Change</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>“SYSUSERAUTHORITY compatibility view (deprecated)” [SQL Anywhere Server - SQL Reference]</td>
<td>This view has changed from a system view to a compatibility view. The underlying ISYSAUTHORITY system table has been removed, but the view is retained for compatibility purposes.</td>
</tr>
<tr>
<td>“SYSTABLEPERM system view” [SQL Anywhere Server - SQL Reference]</td>
<td>New columns: loadauth and truncateauth</td>
</tr>
<tr>
<td>“SYSTABAUTH consolidated view” [SQL Anywhere Server - SQL Reference]</td>
<td>New columns: loadauth and truncateauth</td>
</tr>
</tbody>
</table>

- **Changes to the SQL Anywhere plug-in to support roles and privileges** Sybase Central has undergone many changes to support the new role-based security model. Be aware that performing privileged database tasks—such as creating a table—may require more privileges to perform in Sybase Central than it would using SQL statements. This is because Sybase Central performs additional privileged tasks while you are using it such as populating its folders (Views, Users, and so on) for display.

If you are unsure of the privileges required to perform a task, refer to the documentation for performing the task in Sybase Central.

**LDAP user authentication support**

SQL Anywhere now provides support for LDAP user authentication. The following list provides information on what has been added or changed to support LDAP user authentication. See “LDAP authentication” [SQL Anywhere Server - Database Administration].

- **Changes to the root login policy** The following login policy options have been added to the root login policy to support LDAP user authentication:
  - ldap_primary_server
  - ldap_secondary_server
  - ldap_auto_failback_period
  - ldap_failover_to_std
  - ldap_refresh_dn

See “Root login policy” [SQL Anywhere Server - Database Administration].

- **System procedures** The following system procedures have been added or enhanced to support LDAP user authentication:
  - “sa_get_ldapserver_status system procedure” [SQL Anywhere Server - SQL Reference] (new)
  - “sa_get_user_status system procedure” [SQL Anywhere Server - SQL Reference] (enhanced)

New columns (user_dn, user_dn_cached_at, password_change_state, password_change_first_user, password_change_second_user) are reported by the sa_get_user_status system procedure.
- **Database server and database options** The following database options have been added or enhanced to support LDAP user authentication:
  - “login_mode option” [SQL Anywhere Server - Database Administration]
  - “trusted_certificates_file option” [SQL Anywhere Server - Database Administration]

- **SQL statements** The following SQL statements have been added to support LDAP user authentication:
  - “CREATE LDAP SERVER statement” [SQL Anywhere Server - SQL Reference]
  - “ALTER LDAP SERVER statement” [SQL Anywhere Server - SQL Reference]
  - “DROP LDAP SERVER statement” [SQL Anywhere Server - SQL Reference]
  - “VALIDATE LDAP SERVER statement” [SQL Anywhere Server - SQL Reference]

  The following SQL statements have been enhanced to support LDAP user authentication:
  - “ALTER USER statement” [SQL Anywhere Server - SQL Reference] (new REFRESH DN syntax)
  - “CREATE LOGIN POLICY statement” [SQL Anywhere Server - SQL Reference] (new login policy options)
  - “ALTER LOGIN POLICY statement” [SQL Anywhere Server - SQL Reference] (new login policy options)
  - “COMMENT statement” [SQL Anywhere Server - SQL Reference] (ability to comment on LDAP server configuration objects)

- **Catalog changes** The following catalog changes have been made to support LDAP user authentication:
  - **SYSLDAPSERVER system view (new)** The SYSLDAPSERVER system view contains one row for each LDAP SERVER object configured in the database. See “SYSLDAPSERVER system view” [SQL Anywhere Server - SQL Reference].
  - **SYSUSER system view** The SYSUSER system view has two new columns related to LDAP user authentication: user_dn and user_dn_cached_at. See “SYSUSER system view” [SQL Anywhere Server - SQL Reference].

---

**Product-wide behavior changes**

Following is a list of product-wide behavior changes introduced in version 16.0. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

**Other product-wide behavior changes**

**Sample certificates are located in a new directory** Sample X.509 certificates used for TLS communication have been moved from the *bin32* and *bin64* directories to the *C:\Users\Public\Documents\SQL Anywhere 16\Samples\Certificates* directory.
The following files have also been added to the new directory:

- rsaclient.id
- rsaclient_nopwd.id
- rsaroot.key
- rsaserver_nopwd.id

For more information, see C:\Users\Public\Documents\SQL Anywhere 16\Samples\Certificates\readme.txt.

**SQL Anywhere supports TLS version 1.1**  SQL Anywhere server and client libraries now support TLS version 1.1 for both TLS connection encryption and HTTPS. TLS version 1.1 is not supported on Mac OS X.

**iAnywhere Solutions Oracle ODBC driver renamed**  For version 16.0, the SQL Anywhere Oracle ODBC driver name is SQL Anywhere 16 - Oracle. The previous version was called iAnywhere Solutions 12 - Oracle. For reference, the SQL Anywhere ODBC driver name is SQL Anywhere 16.

### Product-wide deprecated and removed features

- **Elliptic curve encryption (ECC)**  Support for ECC encryption has been removed. This change affects the following features:
  
  - The IsEccAvailable server property has been removed.
  
  - The -ec database server option no longer accepts ECC for the TLS_TYPE protocol option. The TLS_TYPE protocol option for the -ec database server option has been removed from the documentation; however, this option is still supported by the software for backwards compatibility.
  
  - The ENCRYPTION connection parameter no longer accepts ECC for the TLS-TYPE argument. The TLS_TYPE argument for the ENCRYPTION connection has been removed from the documentation; however, this option is still supported by the software for backwards compatibility.
  
  - The Certificate Creation utility (createecert ) no longer accepts the -ec option and the -t option no longer accepts ecc for the encryption type.
  
  - The Key Pair Generator utility (createkey) only creates RSA key pairs. The **Choose encryption type** and **Enter ECC curve** prompts have been removed.
  
  - The SQL Anywhere Monitor does not support ECC encryption. To monitor a pre-version 16 MobiLink server that uses ECC encryption, you must:
    
    - Start the MobiLink server with a second set of network protocol options that does not use ECC.
    
    - Add the MobiLink server as a resource to be monitored. When prompted to specify the port number for the MobiLink server resource, specify the number that is not associated with ECC encryption.
○ The MobiLink tls_type protocol option no longer accepts ECC for the tls_type option. The MobiLink tls_type protocol option has been removed from the documentation; however, this option is still supported by the software.

○ The MobiLink client e2ee_type protocol option no longer accepts ECC for the tls_type option. The MobiLink client e2ee_type protocol option has been removed from the documentation; however, this option is still supported by the software.

○ The Outbound Enabler tls_type option no longer supports ECC as a choice. The tls_type protocol option has been removed from the documentation.

○ UltraLite C/C++ and UltraLiteJ API methods and properties related to ECC configuration have been removed.

● Send column names In MobiLink and UltraLite, column names are now always sent during synchronization. Attempts to disable sending column names are ignored. This change affects the following features:

○ In MobiLink, the SendColumnNames (scn) extended option for dbmlsync has been deprecated and is ignored if set to OFF. The SendColumnNames extended option has been removed from the documentation; however, this option is still supported by the software.

○ UltraLite and UltraLiteJ clients always send column names when synchronizing with a MobiLink server database. Methods and approaches that are used to stop sending column names are now ignored and deprecated. The following members and parameters are affected:

  ● Send Column Names synchronization parameter
  ● Synchronization profile options
  ● SyncParms.setSendColumnNames method [UltraLiteJ]
  ● SyncParms.getSendColumnNames method [UltraLiteJ]
  ● ULSyncParms.SendColumnNames property [UltraLite.NET]
  ● send_column_names in the ul_sync_info structure [UltraLite C and Embedded SQL datatypes]

**SQL Anywhere new features**

Following is a list of new features in SQL Anywhere version 16.0.

For information about changes to the list of supported platforms, see [http://www.sybase.com/detail?id=1061806](http://www.sybase.com/detail?id=1061806).

**Main features**

Following is a list of the main features introduced in SQL Anywhere version 16.0.

**Performance enhancements**

The SQL Anywhere optimizer has been enhanced to consider, during the query optimization process, access plans that are not left-deep trees. Considering extra access plans results in the optimizer finding a
more efficient best plan, which reduces the runtime of some statements significantly in SQL Anywhere 16.0. See “Execution plan components” [SQL Anywhere Server - SQL Usage].

ROW and ARRAY data type support

SQL Anywhere has added support for the ROWS and ARRAYS composite data types. These data types are a more efficient way to store lists because they provide the ability to define the structure and data type of their values. They also make access to list elements easier to achieve, either directly, by using double square brackets; or as result set, by using the UNNEST operator. Consider using the ARRAY data type if you are storing lists as delimited strings in VARCHAR columns, and parsing them using sa_split_list. ARRAYS are very helpful when storing different objects that are all related in some ways. ROWS are helpful when storing multiple values related to one object.

New SQL functions have been added to support composite data types, including a ROW constructor and an ARRAY constructor. A new operator, UNNEST, has also been added

SQL functions and procedures accept ROW and ARRAY data types as IN, OUT, or INOUT arguments. Rows and arrays can be used as the return type from a SQL user-defined function.

ROW and ARRAY types can be used in:

- SELECT, INSERT, UPDATE, MERGE, and DELETE statements
- Transact-SQL and Watcom-SQL stored procedures
- Functions
- Triggers
- Batches

ROW, ARRAY, VARRAY, and UNNEST are now reserved words.

See:

- “Composite data types” [SQL Anywhere Server - SQL Reference]
- “Comparisons of composite types” [SQL Anywhere Server - SQL Reference]
- “Array operators” [SQL Anywhere Server - SQL Reference]
- “ARRAY constructor [Composite]” [SQL Anywhere Server - SQL Reference]
- “ROW constructor [Composite]” [SQL Anywhere Server - SQL Reference]
- “ARRAY_AGG function [Aggregate]” [SQL Anywhere Server - SQL Reference]
- “ARRAY_MAX_CARDINALITY function [Composite]” [SQL Anywhere Server - SQL Reference]
- “CARDINALITY function [Composite]” [SQL Anywhere Server - SQL Reference]
- “TRIM_ARRAY function [Composite]” [SQL Anywhere Server - SQL Reference]
- “UNNEST array operator” [SQL Anywhere Server - SQL Reference]

OData support

SQL Anywhere now includes an OData Server that allows web clients to communicate with a SQL Anywhere database server. Web clients can send OData requests to a configurable OData Producer hosted
in an HTTP server, which converts OData concepts into relational database operations. The following features have been added or modified as part of OData support:

- OData protocol version 2 compliance
- OData Producer
- Embedded Jetty WebServer (with alternative HTTP server support)
- OData Server utility
- OData Server Stop utility
- Service utility (dbsvc) can create OData Windows services
- SQL Anywhere plug-in supports the administration of OData services.

See “OData support” [SQL Anywhere Server - Programming].

**Event tracing support**

Event tracing records information about system-defined and user-defined trace events to an event tracing target. A trace session is made up of trace events (specific points in the database server software or your SQL application) that collect information that is logged to a target. Targets are the location (such as a file) where the database server logs trace events.

Event tracing is recommended for production environments and provides fine-grained control over the information that is logged. You can log both user- and system-defined trace events for both the database server and your application and customize the trace events to identify performance issues.

The trace_system_event secure feature lets you control whether user-defined trace events can be created.

See:

- “Event tracing” [SQL Anywhere Server - Database Administration]
- “sp_trace_events system procedure” [SQL Anywhere Server - SQL Reference]
- “sp_trace_event_fields system procedure” [SQL Anywhere Server - SQL Reference]
- “sp_trace_event_sessions system procedure” [SQL Anywhere Server - SQL Reference]
- “sp_trace_event_session_events system procedure” [SQL Anywhere Server - SQL Reference]
- “sp_trace_event_session_targets system procedure” [SQL Anywhere Server - SQL Reference]
- “sp_trace_event_session_target_options system procedure” [SQL Anywhere Server - SQL Reference]
- “CREATE TEMPORARY TRACE EVENT statement” [SQL Anywhere Server - SQL Reference]
- “CREATE TEMPORARY TRACE EVENT SESSION statement” [SQL Anywhere Server - SQL Reference]
- “ALTER TRACE EVENT SESSION statement” [SQL Anywhere Server - SQL Reference]
- “DROP TRACE EVENT statement” [SQL Anywhere Server - SQL Reference]
- “DROP TRACE EVENT SESSION statement” [SQL Anywhere Server - SQL Reference]
- “NOTIFY TRACE EVENT statement” [SQL Anywhere Server - SQL Reference]
- “Event Trace Data (ETD) File Management utility (dbmanageetd)” [SQL Anywhere Server - Database Administration]
- “Secure features” [SQL Anywhere Server - Database Administration]

**SAP HANA support**

**Interactive SQL supports connecting to SAP HANA databases** You can use Interactive SQL to connect to an SAP HANA database. In the Connect window, click Change Database Type, and then click SAP HANA. See “Interactive SQL” [SQL Anywhere Server - Database Administration].
**HANAODBC remote data access server class** To use SAP HANA as a back-end server, the HANAODBC remote data access class is included in Remote Data Access support. See “Server classes for remote data access” [SQL Anywhere Server - SQL Usage].

**Database mirroring and read-only scale-out enhancements**

**Events can run on the mirror or a copy node** Now, events can run on the mirror server and copy nodes in mirroring and read-only scale-out systems. To create an event that can run on any server, specify the FOR ALL clause with the CREATE EVENT statement or the ALTER EVENT statement. See “CREATE EVENT statement” [SQL Anywhere Server - SQL Reference] and “ALTER EVENT statement” [SQL Anywhere Server - SQL Reference].

**New MIRROR remote data access server class** The MIRROR server class connects to a remote SQL Anywhere server via ODBC. However, when creating the remote server, the USING clause contains a mirror server name from the SYS.SYSMIRRORSERVER catalog table. See “Server class MIRROR” [SQL Anywhere Server - SQL Usage].

**Move the arbiter server in a running mirroring configuration** See “Tutorial: Moving the arbiter server” [SQL Anywhere Server - Database Administration].

**Convert the mirror server to a copy node** See “Tutorial: Converting a partner server to a copy node” [SQL Anywhere Server - Database Administration].

**Move a partner from one server to another server** See “Tutorial: Moving a partner server” [SQL Anywhere Server - Database Administration].

**A server can be a copy node and an arbiter for the same database** See “Tutorial: Using one server as both a copy node and an arbiter” [SQL Anywhere Server - Database Administration].

**Dynamically start a mirroring or read-only scale-out database on a running server** Use the START DATABASE statement with the MIRROR ON clause to start a mirror database or copy node on a running partner. See “START DATABASE statement” [SQL Anywhere Server - SQL Reference].

**SET MIRROR OPTION statement** The SET MIRROR OPTION statement has a new option, promotion_time, that allows you to specify the length of time that a copy node stays connected to the root database server after a parent connection is lost before promoting itself. The max_disconnected_time option now specifies the amount of time since the last time the copy node was connected to the parent, alternate parent, or root database before the copy node stops. See “SET MIRROR OPTION statement” [SQL Anywhere Server - SQL Reference].

**Database property enhancements**

**CopyNodeParent** Returns the name of the current parent server of a copy node in a read-only scale-out configuration. See “CopyNodeParent database property” [SQL Anywhere Server - Database Administration].

**PartnerState** If no partner is defined, then the PartnerState returns NULL. Previously, it returned disconnected. See “PartnerState database property” [SQL Anywhere Server - Database Administration].

**LOAD TABLE new default WITH ROW LOGGING clause** WITH ROW LOGGING clause The WITH ROW LOGGING clause causes each row that is loaded to be recorded in the transaction log as an
INSERT statement. This level of logging is recommended for databases involved in synchronization and is now the default when using the FROM filename-expression or the USING FILE filename-expression on a mirrored database. However, when loading large amounts of data, this logging type can affect performance, and results in a much longer transaction log.

Administration enhancements

- **Databases restart after upgrades**    By default, the database is now stopped and restarted after an upgrade. To stop the database after an upgrade instead of restarting, use the RESTART OFF clause of the ALTER DATABASE statement or the -nrs option of the Upgrade utility (dbupgrad). A database upgrade renames the transaction log once the upgrade completes successfully. See “ALTER DATABASE statement” [SQL Anywhere Server - SQL Reference] and “Upgrade utility (dbupgrad)” [SQL Anywhere Server - Database Administration].

- **Change the owner of a table**       You can now alter the owner of a table by using the ALTER TABLE statement. When altering a table owner, you have the options of preserving or dropping explicitly granted privileges and preserving or dropping foreign keys. See “ALTER TABLE statement” [SQL Anywhere Server - SQL Reference].

- **Dual control passwords**        The change_password_dual_control login policy option requires two administrators to be involved in changing the password for a target user. The first user specifies the first part of a password, the second person specifies the second part. Dual control password changing prevents a password administrator from knowing the complete password of another user.

  The target user must have the change_password_dual_control option enabled in their login policy.

  The function specified by the verify_password_function database option is not called for users that have a login policy with the change_password_dual_control option enabled.

  ○ “Dual control passwords” [SQL Anywhere Server - Database Administration]
  ○ “verify_password_function option” [SQL Anywhere Server - Database Administration]
  ○ “sa_verify_password system procedure” [SQL Anywhere Server - SQL Reference]

Database connections

Following is a list of enhancements made to database connections in SQL Anywhere version 16.0.

- **MaxRequestVars (MAXVARS) protocol option**    This protocol option specifies the maximum number of HTTP input variables allowed in a request that is sent to the database server. See “MaxRequestVars (MAXVARS) protocol option” [SQL Anywhere Server - Database Administration].

- **SuppressInfoForDataTypes ODBC connection parameter**    This connection parameter prevents the ODBC driver from returning information about specified data types. See “ODBC connection parameters” [SQL Anywhere Server - Database Administration].
## Backup and recovery

Following is a list of backup and recovery enhancements introduced in SQL Anywhere version 16.0.

- **Backup utility (dbbackup) enhancements** The following options have been added to the Backup utility. All of these options must be used with the -s option. See “Backup utility (dbbackup)” [SQL Anywhere Server - Database Administration].

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-bc [comment]</td>
<td>Records a comment in the backup history file.</td>
</tr>
<tr>
<td>-wa</td>
<td>Waits after transactions are completed to rename or truncate transaction log.</td>
</tr>
<tr>
<td>-wb</td>
<td>Delays the backup of the database until there are no active transactions.</td>
</tr>
</tbody>
</table>

## Security

Following is a list of security enhancements introduced in SQL Anywhere version 16.0.

- **Raw encryption** The ENCRYPT function can encrypt data inside the database server and output it into a raw format so that it can be decrypted outside of the database. The DECRYPT function can decrypt data encrypted outside of the database server. See “ENCRYPT function [String]” [SQL Anywhere Server - SQL Reference] and “DECRYPT function [String]” [SQL Anywhere Server - SQL Reference].

- **Support for TDS RSA encryption with nonce password exchange** The new -tdsl database server option restricts the type of TDS login request that a database server supports. Set the TDS login mode to support all login requests, only RSA login requests, or only RSA with nonce login requests. See “-tdsl database server option” [SQL Anywhere Server - Database Administration].

- **Secure features**
  - **Secure feature keys can now be customized** You can now create and view customized secure feature keys that can be assigned to individual users by using the following system procedures:
    - **sp_create_secure_feature_key system procedure** Creates a new secure feature key. See “sp_create_secure_feature_key system procedure” [SQL Anywhere Server - SQL Reference].
    - **sp.Alter_secure_feature_key system procedure** Alters a previously defined secure feature key by modifying the authorization key and/or the feature list. See “sp.Alter_secure_feature_key system procedure” [SQL Anywhere Server - SQL Reference].
● **sp_drop_secure_feature_key system procedure**  Deletes a secure feature key. See “sp_drop_secure_feature_key system procedure” [SQL Anywhere Server - SQL Reference].

● **sp_list_secure_feature_keys system procedure**  Returns a list of defined secure feature keys. See “sp_list_secure_feature_keys system procedure” [SQL Anywhere Server - SQL Reference].

● **sp_use_secure_feature_key system procedure**  Allows access to the secured features associated with the specified secure feature key. See “sp_use_secure_feature_key system procedure” [SQL Anywhere Server - SQL Reference].

○ **Secure features**  You can prevent users from manipulating directories and files on the same computer as the server. The following features, which correspond to system procedures, were added to the local_io feature set:

  ● sp_list_directory
  ● sp_create_directory
  ● sp_copy_directory
  ● sp_move_directory
  ● sp_delete_directory
  ● sp_copy_file
  ● sp_move_file
  ● sp_delete_file

The manage_server feature set prevents users from accessing features related to the database server.

The create_trace_file and _trace_system_event features prevent users from creating event tracing targets and user-defined events, respectively.

○ **Re-execute SQL after specifying the database server’s secure feature key**  In Sybase Central, if you try to execute SQL that uses a secure feature, you are given the option of specifying the database server secure feature key, after which the SQL is re-executed. The database server must be started with a secure feature key (using the -sk option) to use this feature.

● **Disk sandboxing**  The disk sandboxing feature limits read-write file operations from the database to the directory where the main database file is located.

When disk sandboxing is enabled, relative path names are treated as relative to the directory where the main database file is located. When disk sandboxing is not enabled, relative path names are relative to the working directory of the database server. See “SQL Anywhere behavior changes” on page 32.

The following features have been added to support the disk sandboxing feature:
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database server options (dbsrv16)</td>
<td>○ <strong>-sbx database server and database option</strong>  This option sets the default disk sandbox behavior for all databases running on the database server or for an individual database. See “-sbx database server option” [SQL Anywhere Server - Database Administration] and “-sbx database option” [SQL Anywhere Server - Database Administration].</td>
</tr>
<tr>
<td>Database options</td>
<td>○ <strong>disk_sandbox database option</strong>  This database option controls whether read-write file operations on the database are restricted to the directory where the main database file is located. See “disk_sandbox option” [SQL Anywhere Server - Database Administration].</td>
</tr>
<tr>
<td>Statements</td>
<td>○ <strong>START DATABASE statement</strong>  Specify the DISKSANDBOX clause when executing the START DATABASE statement to restrict read-write file operations on the database to the directory where the main database file is located. See “START DATABASE statement” [SQL Anywhere Server - SQL Reference].</td>
</tr>
</tbody>
</table>
| Secure features | ○ **manage_disk_sandbox secure feature**  Use the -sf database server option to control the manage_disk_sandbox secure feature. The manage_disk_sandbox secure feature prevents users from changing disk sandbox settings. By default, the manage_disk_sandbox secure feature is secured. See “-sf database server option” [SQL Anywhere Server - Database Administration].  

 ○ **disk_sandbox secure feature**  Use the -sf database server option to control the disk_sandbox secure feature. The disk_sandbox secure feature allows connected users to enable or disable disk sandboxing for their connection. By default, the disk_sandbox secure feature is secured. See “Disk sandboxing” [SQL Anywhere Server - Database Administration]. |
| System procedures | To use either the sa_server_option system procedure or the sa_db_option system procedure to change disk sandbox settings, provide the secure feature key for the manage_disk_sandbox secure feature.  

 ○ **sa_server_option system procedure**  Changes the default disk sandbox behavior for the database server while the database server is running. See “sa_server_option system procedure” [SQL Anywhere Server - SQL Reference].  

 ○ **sa_db_option system procedure**  Changes the disk sandbox behavior for the database while the database is running. See “sa_db_option system procedure” [SQL Anywhere Server - SQL Reference]. |
**Database utilities**

Following is a list of enhancements made to database utilities in SQL Anywhere version 16.0.

- **Start Server in Background utility (dbspawn)**  Using dbspawn to spawn a database server with a name that is not unique returns a new error and exit code (EXIT_SERVER_NAME_IN_USE). See “Software component exit codes” [SQL Anywhere Server - Programming].

- **Unload utility (dbunload)**  The -ss option allows you to suppress the generation of column statistics in the reload SQL script file. See “Unload utility (dbunload)” [SQL Anywhere Server - Database Administration].

- **Certificate Creation utility (createcert) and Certificate Viewer utility (viewcert)**
  - The -p1 option for createcert and viewcert allows you to specify the use of PKCS #1 to encode unencrypted RSA private keys.
  - The -sa option for createcert allows you to specify the signature algorithm to use when creating or signing certificates.
  - The @data option for viewcert allows you to run viewcert the command-line specifications stored in a configuration file or environment variable.

  See “Certificate Creation utility (createcert)” [SQL Anywhere Server - Database Administration] and “Certificate Viewer utility (viewcert)” [SQL Anywhere Server - Database Administration].

**Database options**

Following is a list of enhancements made to database options in SQL Anywhere version 16.0.

- **auto_commit_on_create_local_temp_index option**  This option controls whether the database server performs a COMMIT before an index is created on a local temporary table. You must upgrade existing databases to use this feature. See “auto_commit_on_create_local_temp_index option” [SQL Anywhere Server - Database Administration].
• **extern_login_credentials option** Controls whether remote connections are be attempted using the logged in user's extern login credentials or the effective user's extern login credentials. See “extern_login_credentials option” [SQL Anywhere Server - Database Administration].

• **webservice_sessionid_name option** This option redefines what the SQL Anywhere web server uses to determine whether session management is used. See “webservice_sessionid_name option” [SQL Anywhere Server - Database Administration].

## Database server options

Following is a list of enhancements made to database server options in SQL Anywhere version 16.0.

- **-al database server option** Allows standard user authentication for specified users. See “-al database server option” [SQL Anywhere Server - Database Administration].

- **-al database option** Allows standard user authentication for specified users of the specified database. See “-al database option” [SQL Anywhere Server - Database Administration].

- **-gta database server option** Sets which logical processors the database server can use. See “-gta database server option” [SQL Anywhere Server - Database Administration].

- **-uf database server option** The -uf database server option now applies to all operating systems. See “-uf database server option” [SQL Anywhere Server - Database Administration].

- **-ufd database server option** Specifies the action that the database server takes when a fatal error or assertion failure occurs on a database. See “-ufd database server option” [SQL Anywhere Server - Database Administration].

- **-xs database server option** The -xs database server option can be specified multiple times to allow the database server to listen for connections on multiple ports and/or protocols. Previously the -xs database server option could only be specified once, so you had to specify all the ports and protocols within the parameter of the single -xs option. See “-xs database server option” [SQL Anywhere Server - Database Administration].

## Properties and Performance Monitor statistics

Following is a list of enhancements made to properties and Performance Monitor statistics in SQL Anywhere version 16.0.
● **New connection properties**  The following connection properties have been added in this release:

  ○ “auto_commit_on_create_local_temp_index connection property” [SQL Anywhere Server - Database Administration]
  ○ “extern_login_credentials connection property” [SQL Anywhere Server - Database Administration]
  ○ “NumLocalTempTables connection property” [SQL Anywhere Server - Database Administration]
  ○ “trusted_certificates_file connection property” [SQL Anywhere Server - Database Administration]
  ○ “webservice_sessionid_name connection property” [SQL Anywhere Server - Database Administration]

● **New database properties**  The following database properties have been added in this release:

  ○ LastCommitRedoPos
  ○ LastSyncedRedoPos
  ○ LastWrittenRedoPos
  ○ UTCTimestampCatalog

● **New database server properties**  The following database server properties have been added in this release:

  ○ IsAesniAvailable
  ○ ProcessorAffinity

### System procedures and functions

Following is a list of system procedure and function enhancements added in SQL Anywhere version 16.0.

● **sa_certificate_info system procedure**  This system procedure returns information about certificates stored in the database. See “sa_certificate_info system procedure” [SQL Anywhere Server - SQL Reference].

● **sa_cpu_topology system procedure**  This system procedure returns information about the processor topology of the computer that the database server is running on. See “sa_cpu_topology system procedure” [SQL Anywhere Server - SQL Reference].

● **sa_db_option system procedure**  This system procedure allows you to override a database option while the database is running. See “sa_db_option system procedure” [SQL Anywhere Server - SQL Reference].

● **sa_parse_json system procedure**  This system procedure returns a representation of JSON data using ROW and ARRAY SQL types. See “sp_parse_json system procedure” [SQL Anywhere Server - SQL Reference].

● **sa_server_option system procedure**  On Windows and Linux, you can change the number of logical processors the database server can use after the database server has been started by using the ProcessorAffinity property of the sa_server_option system procedure. See “sa_server_option system procedure” [SQL Anywhere Server - SQL Reference].
• **Directory and file stored procedures**  A new set of stored procedures allow you to list and manipulate directories and files that are accessible by the server without requiring the need to set up directory access servers, external logins, and directory access proxy tables. These system procedures can be enabled and disabled as secure features.

  ○ **sp_copy_directory system procedure**  This system procedure copies a directory. See “sp_copy_directory system procedure” [SQL Anywhere Server - SQL Reference].

  ○ **sp_copy_file system procedure**  This system procedure copies a file. See “sp_copy_file system procedure” [SQL Anywhere Server - SQL Reference].

  ○ **sp_create_directory system procedure**  This system procedure creates a directory. See “sp_create_directory system procedure” [SQL Anywhere Server - SQL Reference].

  ○ **sp_delete_directory system procedure**  This system procedure deletes a directory. See “sp_delete_directory system procedure” [SQL Anywhere Server - SQL Reference].

  ○ **sp_delete_file system procedure**  This system procedure deletes a file. See “sp_delete_file system procedure” [SQL Anywhere Server - SQL Reference].

  ○ **sp_move_directory system procedure**  This system procedure moves a directory. See “sp_move_directory system procedure” [SQL Anywhere Server - SQL Reference].

  ○ **sp_move_file system procedure**  This system procedure moves a file. See “sp_move_file system procedure” [SQL Anywhere Server - SQL Reference].

• **xp_get_mail_error_code system procedure**  This system procedure returns information about the most recent SMTP or MAPI error when using the xp_mail procedures. See “xp_get_mail_error_code system procedure” [SQL Anywhere Server - SQL Reference].

• **xp_get_mail_error_text system procedure**  This system procedure returns information about the most recent SMTP error when using the xp_mail procedures. See “xp_get_mail_error_text system procedure” [SQL Anywhere Server - SQL Reference].

• **xp_getenv system procedure**  This system procedure returns the value of an environment variable. See “xp_getenv system procedure” [SQL Anywhere Server - SQL Reference].

• **xp_startsmtpl system procedure**  The trusted_certificates parameter of this system procedure can now contain `file=file-path` and `cert_name=certificate-name` options. See “xp_startsmtpl system procedure” [SQL Anywhere Server - SQL Reference].

• **Subselect and subqueries supported for parameter expressions in functions and procedures.**  Subselect and subqueries are supported for parameter expressions in functions and procedures.

• **BINTOHEX function**  This function returns the hexadecimal equivalent of a binary string. See “BINTOHEX function [Data type conversion]” [SQL Anywhere Server - SQL Reference].

• **HEXTOBIN function**  This function returns the LONG BINARY equivalent of a hexadecimal string. See “HEXTOBIN function [Data type conversion]” [SQL Anywhere Server - SQL Reference].
SQL statements

Following is a list of new and enhanced SQL statements introduced in SQL Anywhere version 16.0.

**ALTER TABLE statement** You can now alter the owner of a table. When altering a table owner, you have the options of preserving or dropping owner permissions and preserving or dropping foreign keys. See “ALTER TABLE statement” [SQL Anywhere Server - SQL Reference].

**CREATE CERTIFICATE and DROP CERTIFICATE statements** You can now store certificates in the database. The CREATE CERTIFICATE statement can be used to add or replace certificates to a database. The DROP CERTIFICATE statement can be used to remove certificates from a database. See “CREATE CERTIFICATE statement” [SQL Anywhere Server - SQL Reference] and “DROP CERTIFICATE statement” [SQL Anywhere Server - SQL Reference].

**CREATE EVENT and ALTER EVENT statements** Now, events can run on mirror servers and copy nodes in mirroring and read-only scale-out systems. To create an event that can run on any server, specify the FOR ALL clause with the CREATE EVENT statement or the ALTER EVENT statement. See “CREATE EVENT statement” [SQL Anywhere Server - SQL Reference] and “ALTER EVENT statement” [SQL Anywhere Server - SQL Reference].

**CREATE FUNCTION statement [Web service]** You can now specify a certificate stored in the database using the certificate_name option of the CERTIFICATE clause. You can also now specify a keep-alive timeout criteria option (kto) to instantiate and cache a keep-alive HTTP/HTTPS connection for a period of time. See “CREATE FUNCTION statement [Web service]” [SQL Anywhere Server - SQL Reference].

**CREATE INDEX statement** You can now specify WITH NULLS DISTINCT when executing the statement. See “CREATE INDEX statement” [SQL Anywhere Server - SQL Reference].

**CREATE PROCEDURE statement [Web service]** You can now specify a certificate stored in the database using the certificate_name option of the CERTIFICATE clause. You can also now specify a keep-alive timeout criteria option (kto) to instantiate and cache a keep-alive HTTP/HTTPS connection for a period of time. See “CREATE PROCEDURE statement [Web service]” [SQL Anywhere Server - SQL Reference].

**CREATE SERVER statement** The USING clause DELIMITER option is used to force the use of either a slash (/) or backslash (\) as a path separator in the file names returned by the directory access server. See “CREATE SERVER statement” [SQL Anywhere Server - SQL Reference].

**CREATE TEXT INDEX statement** You can now create text indexes on materialized views. Use the IMMEDIATE REFRESH clause to create a text index on a materialized view. See “CREATE TEXT INDEX statement” [SQL Anywhere Server - SQL Reference].

**DROP REMOTE CONNECTION statement** The DROP REMOTE CONNECTION statement drops remote data access connections to a remote server. See “DROP REMOTE CONNECTION statement” [SQL Anywhere Server - SQL Reference].

**CREATE TEMPORARY TRACE EVENT SESSION statement** Creates a user trace event session. See “CREATE TEMPORARY TRACE EVENT SESSION statement” [SQL Anywhere Server - SQL Reference].
CREATE TEMPORARY TRACE EVENT  Creates a user trace event that persists until the database is stopped. See “CREATE TEMPORARY TRACE EVENT statement” [SQL Anywhere Server - SQL Reference].

ALTER TRACE EVENT SESSION statement  Adds or removes trace events from a session, adds or removes targets from a session, or starts or stops a trace session. See “ALTER TRACE EVENT SESSION statement” [SQL Anywhere Server - SQL Reference].

DROP TRACE EVENT statement  Drops a user-defined trace event. See “DROP TRACE EVENT statement” [SQL Anywhere Server - SQL Reference].

DROP TRACE EVENT SESSION statement  Drops a trace event session. See “DROP TRACE EVENT SESSION statement” [SQL Anywhere Server - SQL Reference].

LOAD TABLE statement  For the load-option clause, you can now specify the following:
  ○ ALLOW ERRORS clause  Allows you to specify the number of errors allowed before the load operation rolls back. The default is 0.
  ○ ROW LOG clause  If an error is encountered while inserting or parsing a row, the database server writes an image of the input row to the specified location in addition to reporting the row to the user.
  ○ MESSAGE LOG clause  When an error is encountered while inserting or parsing a row, the database server writes the error to the specified location.

NOTIFY TRACE EVENT statement  Logs a user-defined trace event to a trace session. See “NOTIFY TRACE EVENT statement” [SQL Anywhere Server - SQL Reference].

TRY...CATCH statement  You can use a BEGIN TRY...END TRY statement (with BEGIN CATCH...END CATCH) as an error handler to obtain information about errors that occur within the compound statements. See “TRY statement” [SQL Anywhere Server - SQL Reference].

The following functions and system procedures have been added to obtain information about errors. The procedures and functions can be used anywhere within a statement, not just within an error handler.
  ○ “ERROR_LINE function [Miscellaneous]” [SQL Anywhere Server - SQL Reference]
  ○ “ERROR_MESSAGE function [Miscellaneous]” [SQL Anywhere Server - SQL Reference]
  ○ “ERROR_PROCEDURE function [function type]” [SQL Anywhere Server - SQL Reference]
  ○ “ERROR_SQLCODE function [Miscellaneous]” [SQL Anywhere Server - SQL Reference]
  ○ “ERROR_SQLSTATE function [Miscellaneous]” [SQL Anywhere Server - SQL Reference]
  ○ “ERROR_STACK_TRACE function [Miscellaneous]” [SQL Anywhere Server - SQL Reference]
  ○ “sa_error_stack_trace system procedure” [SQL Anywhere Server - SQL Reference]
  ○ “sa_stack_trace system procedure” [SQL Anywhere Server - SQL Reference]
  ○ “STACK_TRACE function [Miscellaneous]” [SQL Anywhere Server - SQL Reference]
Statements that include an ENCRYPTED KEY or KEY clause can now use variable names. You can specify either a string or a variable name for the ENCRYPTED KEY or KEY clause. This change affects the following statements:

- ALTER DATABASE statement
- CREATE DATABASE statement
- CREATE DECRYPTED DATABASE statement
- CREATE DECRYPTED FILE statement
- CREATE ENCRYPTED DATABASE statement
- CREATE ENCRYPTED FILE statement
- DROP DATABASE statement
- LOAD TABLE statement
- RESTORE DATABASE statement
- START DATABASE statement
- UNLOAD statement
- Openstring expressions in a FROM clause

Data types
Following is a list of enhancements to data types introduced in SQL Anywhere version 16.0.

- **Support added for trailing UNSIGNED keyword** The database server now provides support for BIGINT UNSIGNED, INTEGER UNSIGNED, SMALLINT UNSIGNED, and TINYINT UNSIGNED type declarations. See “Numeric data types”.

Programming interfaces
Following is a list of enhancements to programming interfaces introduced in SQL Anywhere version 16.0.

- **Support added for Visual Studio 2012** The SQL Anywhere .NET Data Provider now provides support for Microsoft Visual Studio 2012. See “SQL Anywhere .NET support”.

- **Provider support added for .NET Compact Framework 3.5** The SQL Anywhere .NET Data Provider now provides support for the Compact Framework 3.5. See “SQL Anywhere .NET Data Provider features”.

- **Provider support added for .NET Framework 4.5** The SQL Anywhere .NET 4.0 Data Provider now provides support for .NET Framework 4.5. See “SQL Anywhere .NET Data Provider features”.

- **Provider support added for Entity Framework 5.0** The SQL Anywhere .NET Data Provider now provides support for Entity Framework 5.0, the latest release of Microsoft Entity Framework. See “Entity Framework support”.

- **OLE DB provider now supports the DBTYPE_DBTIME2 data type** The SQL Anywhere OLE DB provider now supports the DBTYPE_DBTIME2 data type. DBTYPE_DBTIME2 (145) is an OLE...
DB type that supports the TIME data type with fractional seconds (DBTYPE_DBTIME does not support fractional seconds). Support for this data type facilitates transfer of columns of type TIME between SQL Anywhere databases and other database management systems (including SQL Anywhere) without loss of precision.

- **Updated PHP driver support**  Support has been added for newer PHP versions up to and including 5.3.18 and 5.4.8.

## Catalog changes

Following is a list of catalog changes introduced in SQL Anywhere version 16.0.

- **New ISYSCERTIFICATE system table and SYSCERTIFICATE system view**  Each row in the ISYSCERTIFICATE system table stores a certificate in text PEM-format. The SYSCERTIFICATE system view is a view on this system table. See “ISYSCERTIFICATE system table” [SQL Anywhere Server - SQL Reference] and “SYSCERTIFICATE system view” [SQL Anywhere Server - SQL Reference].

- **New columns in ISYSTABCOL and ISYSTAB**  The columns nonmaterialized_value and start_schema have been added to ISYSTABCOL. The column current_schema has been added to ISYSTAB. See “SYSTAB system view” [SQL Anywhere Server - SQL Reference] and “SYSTABCOL system view” [SQL Anywhere Server - SQL Reference].

- **New columns in ISYSUSER**  The columns user_type, user_dn, user_dn_cached_at, password_creation_time_utc, last_login_time_utc, and dual_password have been added to ISYSUSER. See “SYSUSER system view” [SQL Anywhere Server - SQL Reference].

- **UTC timestamps used in the catalog**  In previous releases, catalog tables stored and exposed local timestamps that depended on the location of the database server. Timestamps are now stored as their UTC (Coordinated Universal Time) based equivalent, but views expose both UTC timestamps and local timestamps. Applications that rely on the number of columns in the following system tables and views must be updated:

<table>
<thead>
<tr>
<th>System table name</th>
<th>System view name</th>
<th>Column(s) added</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISYSTAB</td>
<td>SYSTAB</td>
<td>last_modified_at_utc</td>
</tr>
<tr>
<td>ISYSVIEW</td>
<td>SYSVIEW</td>
<td>mv_lastrefreshed_at_utc,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mv_known_stale_at_utc</td>
</tr>
<tr>
<td>ISYSOBJECT</td>
<td>SYSOBJECT</td>
<td>creation_time_utc</td>
</tr>
<tr>
<td>ISYSUSER</td>
<td>SYSUSER</td>
<td>password_creation_time,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>last_login_time</td>
</tr>
<tr>
<td>ISYSTEXTIDX</td>
<td>SYSTEXTIDX</td>
<td>last_refresh_utc</td>
</tr>
</tbody>
</table>
Unix/Linux enhancements

Following is a list of Unix and Linux enhancements introduced in SQL Anywhere version 16.0.

- **SQL Anywhere localized French resources for Linux**  Localized French resources for SQL Anywhere are now available on Linux. Localization affects many components, including packaging, installation, documentation, software user interface, and error/warning/information messages. See “Localized versions of SQL Anywhere” [SQL Anywhere Server - Database Administration].

Performance enhancements

Following is a list of performance enhancements introduced in version 16.0 for which there are no user-visible changes other than performance improvement.

- **AES encryption**  Performance has been improved when using AES to encrypt client/server communications and to encrypt database files on Intel/AMD processors for Windows and Linux. You can check if an enhanced implementation of the AES algorithm is being used by querying the IsAesniAvailable database server property. See “IsAesniAvailable server property” [SQL Anywhere Server - Database Administration].

- **Adding columns to tables**  Performance has been improved when using the ALTER TABLE...ADD COLUMN statement. In most situations where the default value can be evaluated to a constant, the ALTER TABLE statement should run significantly faster than in previous releases. You must upgrade existing version 11 or later databases to be able to use this feature. You must rebuild version 10 and earlier databases to use this feature.
**Miscellaneous**

Following is a list of miscellaneous enhancements introduced in SQL Anywhere version 16.0.

- **New and enhanced expressions for the CONTAINS search condition** A new before-expression has been added to the CONTAINS search condition. Use the before-expression to perform a proximity search that respects the relative order of terms. See “CONTAINS search condition” [SQL Anywhere Server - SQL Reference].

  Additionally, the near-expression has been enhanced to accept minimum distance conditions between search terms. See “CONTAINS search condition” [SQL Anywhere Server - SQL Reference].

- **Client messages about encryption** When a database is created using the dbinit utility or the CREATE DATABASE statement, messages are now sent to the client indicating what type of database encryption is used. If encryption is used, the algorithm being used is also displayed.

- **Graphical and long plans** Graphical and long plans have been enhanced to contain optimization statistics collected during the query optimization process. See “Graphical plans” [SQL Anywhere Server - SQL Usage] and “Long text plan” [SQL Anywhere Server - SQL Usage].

- **CESU-8 support** SQL Anywhere now supports the CESU-8 character set (its alias is ibm-9400) and the CESU8BIN collation. Use the dbinit utility with the -le+ option to obtain a complete list of supported character sets. For information about supported collations, see “Alternate collations” [SQL Anywhere Server - Database Administration].

- **Reserved words** The following is a list of reserved words added to the database in SQL Anywhere version 16.0. See “Reserved words” [SQL Anywhere Server - SQL Reference].

  - array
  - json
  - row
  - rowtype
  - unnest
  - varray

- **Identifiers and aliases** Identifiers and aliases can no longer contain the following characters:

  - Square brackets
  - Back quotes

**Note**

If you are reloading a database that is of an earlier version than 16.0, remove any square brackets or back quotes in identifiers; otherwise, the reload fails.

For more information about character restrictions for identifiers, see “Identifiers” [SQL Anywhere Server - SQL Reference]. For more information about character restrictions for aliases, see the select-list clause in the “SELECT statement” [SQL Anywhere Server - SQL Reference].
**Named parameters**  SQL Anywhere now supports the SQL/2011 named parameter syntax. The following named parameter syntax is new:

- `parameter-name => parameter-value`

Functions and procedures that are referenced from the CALL statement, the EXECUTE statement (Transact-SQL), the FROM clause of a DML statement, and the TRIGGER EVENT statement support named parameter syntax. Named parameters allow you to specify any subset of the available parameters in any order. For more information, see the “Named parameters” [SQL Anywhere Server - SQL Reference](#).

---

**SQL Anywhere behavior changes**

Following is a list of behavior changes to SQL Anywhere introduced in version 16.0. For information about supported platforms and versions, see [http://www.sybase.com/detail?id=1061806](http://www.sybase.com/detail?id=1061806).

- **Default security model for some system procedures has changed**  Some pre-16.0 system procedures perform operations in the database that required permissions. In the pre-16.0 security model, these procedures executed with the permissions of the definer (owner), so that a user only needed permission to execute the procedure itself, not the permissions for all the operations the procedure would perform. There were some exceptions where DBA authority was also required.

  As of version 16.0 and higher, the default security model changes for these procedures. By default in newly created databases, these procedures execute with the privileges of the person invoking them. So, to execute a procedure, the invoker needs to have the privileges noted in the documentation for the procedure. The invoker also needs EXECUTE privilege on the procedure, but they inherit this by being a member of PUBLIC.

  You can control whether to use the old model or new model at database creation time using the new SYSTEM PROCEDURE AS DEFINER clause of the CREATE DATABASE statement, or the -pd option of the Initialization Utility (dbinit). However, some system procedures always require the privileges noted in the documentation, regardless of the security model setting. For a list of these procedures, see “Running pre-16.0 system procedures as invoker or definer” [SQL Anywhere Server - SQL Usage](#).

  When upgrading a database, the default behavior is to maintain the security model that is already in place. So, if you are upgrading a 12.0.1 database, for example, your upgraded database will use the old security model, unless you specify otherwise.

  To control which security model to use at upgrade time, use the SYSTEM PROCEDURE AS DEFINER clause of the ALTER DATABASE statement, or the -pd option of the Upgrade Utility (dbupgrad). The same exceptions regarding the set of system procedures that always need both EXECUTE and specific privileges apply for upgrading as for creation.
Note
The security model decision (invoker vs. definer) does not impact the default behavior for user-defined procedures, which continues to default to definer. Even if the default for system procedures is changed to invoker, the default for user-defined procedures remains as definer.

○ “Changes for system procedures that perform privileged operations” [SQL Anywhere Server - Database Administration]
○ “Running pre-16.0 system procedures as invoker or definer” [SQL Anywhere Server - SQL Usage]
○ “List of procedures that are impacted by the invoker/definer setting” [SQL Anywhere Server - SQL Usage]
○ “ALTER DATABASE statement” [SQL Anywhere Server - SQL Reference]
○ “CREATE DATABASE statement” [SQL Anywhere Server - SQL Reference]
○ “Initialization utility (dbinit)” [SQL Anywhere Server - Database Administration]
○ “Upgrade utility (dbupgrad)” [SQL Anywhere Server - Database Administration]

● Relative paths and disk sandboxing   In previous releases, relative paths always defaulted to the working directory of the database server. Now, if disk sandboxing is enabled for a database, relative paths are relative to the database directory, not the database server directory. See “Disk sandboxing” [SQL Anywhere Server - Database Administration].

● SELECT * supported in view definitions   In previous releases, SELECT * was only supported in the main query of the CREATE VIEW statement. Now it is supported in the main query, a subquery, a derived table, or a subselect of the CREATE VIEW statement. See “CREATE VIEW statement” [SQL Anywhere Server - SQL Reference].

● Minimum password length changed   In previous releases, the default minimum password length was 0 characters. The default minimum length for passwords has been changed to 3 characters. If your application allows passwords that are 0 characters, you can execute the following statement on new SQL Anywhere databases to change the default setting to match previous releases:

```sql
SET OPTION PUBLIC.min_password_length=0;
```

For more information, see “min_password_length option” [SQL Anywhere Server - Database Administration].

● Default behavior has changed when creating indexes on local temporary tables   In previous releases, the database server always executed a COMMIT before creating an index on a local temporary table. Now, the database server does not perform a COMMIT before creating an index on a local temporary table. You can control this behavior by setting the auto_commit_on_create_local_temp_index database option. See “auto_commit_on_create_local_temp_index option” [SQL Anywhere Server - Database Administration].

● New maximum packet size   The maximum packet size has increased from 16000 to 65535 bytes. SQL Anywhere 12 and earlier clients are limited to 16000 bytes when they are connected to SQL Anywhere 16 database servers. The default packet size has not changed. See “CommBufferSize (CBSIZE) connection parameter” [SQL Anywhere Server - Database Administration].

● -ch database server option limit changed   If you specify a maximum cache size that is less than 64 MB with the -ch option, the database server adjusts the maximum cache size to 64 MB. If you
require a maximum cache size that is less than 64 MB (this setting is not recommended), then you can use the -chx option. These changes do not apply on Windows Mobile. See “-ch database server option” [SQL Anywhere Server - Database Administration].

- **Concurrent index building**  In previous releases, the CREATE INDEX statement acquired an EXCLUSIVE table lock when an index was being built. Now, the operation acquires an EXCLUSIVE table lock for short periods of time at the beginning and at the end of the operation and a SHARED lock for most of the operation, so that other connections can access the table data while the index is being created. The connection creating the index is blocked from accessing the table until the index is created. You must upgrade existing databases to use this feature. See “CREATE INDEX statement” [SQL Anywhere Server - SQL Reference].

- **Mirroring connections can dropped when the connections prevent the transaction log from being applied**  Connections to a copy-node or the mirror database are dropped in some cases when these connections prevent the transaction log from being applied. For example, if a connection is using a procedure that the transaction log is trying to alter or drop, then the connection that is blocking the transaction log from being applied is dropped, and a message is printed to the server console. See “Queries executed on the mirror database” [SQL Anywhere Server - Database Administration].

- **Plan caching changes**  Query execution plans are not cached for queries that have long running times because the benefits of avoiding query optimization are small compared to the total cost of the query. Additionally, the database server does not try to reconstruct reusable query plans for queries that are very sensitive to the values of their host variables.

- **New default for request variables to HTTP server**  In previous releases, there was no limit on the number of HTTP input variables that could be sent in a request. The MaxRequestVars protocol option limits the number of HTTP input variables. The default for the MaxRequestVars protocol option is 10000 variables. See “MaxRequestVars (MAXVARS) protocol option” [SQL Anywhere Server - Database Administration].

- **New default behavior for database assertions**  In previous releases, database assertion failures were treated as database server assertion failures, causing the database server to shut down or return an error to all client connections. Database assertion failures are now treated separately from database server assertion failures, and cause the database to shut down while the database server continues to run. See “-ufd database server option” [SQL Anywhere Server - Database Administration].

- **Secure feature behavior changes**  Secure feature keys now have a minimum length of 6 characters and some secure features cannot be disabled globally—they can only be disable for individual connections. See “Secure features” [SQL Anywhere Server - Database Administration].

- **Sensitive information is obfuscated in output**  Passwords and encryption keys are obfuscated when statements that contain them are printed to the request log, logged by diagnostic tracing, or used as column names by DESCRIBE statements. This behavior also applies to output of the REWRITE function, the LastStatement connection property, and statements recorded in event tracing output. Sensitive parameters and passwords and keys are hidden for the following functions, procedures, and statements:
- ENCRYPT function
- DECRYPT function
- sa_verify_password system procedure
- sp_addlogin procedure (Adaptive Server Enterprise compatibility procedure)
- sp_password system procedure
- xp_startmail system procedure
- xp_startsmtp system procedure
- sp_create_secure_feature_key system procedure
- sp_alter_secure_feature_key system procedure
- sp_use_secure_feature_key system procedure
- GRANT CONNECT statement
- CREATE DATABASE statement
- CREATE ENCRYPTED DATABASE statement
- CREATE ENCRYPTED TABLE DATABASE statement
- CREATE ENCRYPTED FILE statement
- CREATE DECRYPTED DATABASE statement
- CREATE DECRYPTED FILE statement
- START DATABASE statement
- DROP DATABASE statement
- CREATE EXTERNLOGIN statement
- SET TEMPORARY OPTION secure_feature_key = key statement

- **Personal server (dbeng16) licensing change**  The personal server is limited to four cores on one CPU. Previously, the personal server was limited to one CPU. See “Server Licensing utility (dblic)” [SQL Anywhere Server - Database Administration].

### Database utility behavior changes

- **Upgrade utility (dbupgrad) behavior change**  By default, the database is restarted after an upgrade. To prevent the database from restarting, use the -nrs option to stop the database after the upgrade. When an upgrade completes successfully, the transaction log is renamed. See “Upgrade utility (dbupgrad)” [SQL Anywhere Server - Database Administration].

- **Support utility (dbsupport) behavior change**
  - **-cc autosubmit option**  Previously, the -cc autosubmit option automatically submitted crash reports and diagnostic information only when a crash occurred. Now, the -cc autosubmit option automatically submits crash reports and regularly submits diagnostic information to the software development team to help improve the product.

    During installation, this option is enabled by default. See “Regularly submit performance data” [SQL Anywhere Server - Database Administration]

  - **-cp autodetect option**  The -cp autodetect option is now supported on Unix platforms. On Unix, the -cp autodetect option allows dbsupport to configure its proxy server and port by using the proxy server and port set by the HTTP_PROXY environment variable. See “Support utility (dbsupport)” [SQL Anywhere Server - Database Administration].
- **-ch option (deprecated)** The -ch and -ch- options of dbsupport are no longer supported. Additionally, the -e option of dbsupport no longer accepts the -ch option. That is, you cannot specify -ech. See “Support utility (dbsupport)” [SQL Anywhere Server - Database Administration].

### System procedure and function behavior changes

Following is a list of system procedure and function behavior changes added in SQL Anywhere version 16.0.

- **Returning information for other connections now requires privileges** For version 12 or earlier databases, you do not need any permissions to execute the following system procedures and functions to return information for any connection. For version 16 databases, any user can execute these system procedures and functions to return information for the current connection. However, to execute these system procedures and functions to return information for other connections, you must have either the SERVER OPERATOR, MONITOR, or DROP CONNECTION system privilege.

  - “CONNECTION_EXTENDED_PROPERTY function [String]” [SQL Anywhere Server - SQL Reference]
  - “CONNECTION_PROPERTY function [System]” [SQL Anywhere Server - SQL Reference]
  - “NEXT_CONNECTION function [System]” [SQL Anywhere Server - SQL Reference]
  - “sa_conn_activity system procedure” [SQL Anywhere Server - SQL Reference]
  - “sa_conn_compression_info system procedure” [SQL Anywhere Server - SQL Reference]
  - “sa_conn_info system procedure” [SQL Anywhere Server - SQL Reference]
  - “sa_conn_list system procedure” [SQL Anywhere Server - SQL Reference]
  - “sa_conn_properties system procedure” [SQL Anywhere Server - SQL Reference]

- **Returning information for other databases now requires privileges** For version 12 and earlier databases, you did not need any permissions to execute the following system procedures and functions to return information for any database. For version 16 databases, any user can execute these system procedures and functions to return information for the current database. However, to execute these system procedures and functions to return information for other databases, you must have either the SERVER OPERATOR or MONITOR system privilege.

  - “DB_EXTENDED_PROPERTY function [System]” [SQL Anywhere Server - SQL Reference]
  - “DB_NAME function [System]” [SQL Anywhere Server - SQL Reference]
  - “DB_PROPERTY function [System]” [SQL Anywhere Server - SQL Reference]
  - “NEXT_DATABASE function [System]” [SQL Anywhere Server - SQL Reference]
  - “sa_db_info system procedure” [SQL Anywhere Server - SQL Reference]
  - “sa_db_list system procedure” [SQL Anywhere Server - SQL Reference]
  - “sa_db_properties system procedure” [SQL Anywhere Server - SQL Reference]

- **TRACEBACK function enhancements** The TRACEBACK function returns a call stack that is annotated with the object names and line numbers, which simplifies the process of locating the actual
statements that are reported in the call stack. See “TRACEBACK function [Miscellaneous]” [SQL Anywhere Server - SQL Reference].

- **Changes to SMTP and MAPI return codes**  The following changes have been made to the return codes used by the xp_startmail, xp_startsmtp, xp_sendmail, xp_stopmail, and xp_stopssmtp system procedures:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning in previous releases</th>
<th>Meaning in version 16.0 databases</th>
<th>Additional information</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>Unknown error</td>
<td>New error code</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Success</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>An invalid parameter was supplied</td>
<td>New error code</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>xp_startmail or xp_startsmtp failed</td>
<td>Out of memory</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>xp_stopmail or xp_stopssmtp failed</td>
<td>xp_startmail or xp_startsmtp was not called</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Bad host name</td>
<td>New error code</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>xp_sendmail failed</td>
<td>Connect error</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Secure connection error</td>
<td>New error code</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MAPI functions are not available</td>
<td>New error code</td>
<td></td>
</tr>
</tbody>
</table>

The xp_get_mail_error_code and xp_get_mail_error_text system procedures return additional information about the return codes. You must upgrade existing databases to get the new system procedures and error codes. See:

- “xp_get_mail_error_code system procedure” [SQL Anywhere Server - SQL Reference]
- “xp_get_mail_error_text system procedure” [SQL Anywhere Server - SQL Reference]
- “Return codes for MAPI and SMTP system procedures” [SQL Anywhere Server - SQL Reference]

If your application cannot be changed to check for the updated return codes, you can configure the database server to use the return codes from previous releases of the software by using the version 12 dbext.dll file and the use_old_dbextf.sql script, located in the scripts subdirectory of your SQL Anywhere installation. See “Database server deployment” [SQL Anywhere Server - Programming].

- **USER_NAME and SUSER_NAME functions now return VARCHAR**  The user_name and suser_name functions now return data of type VARCHAR instead of LONG VARCHAR. If you have materialized views that use these functions, you need to rebuild them.

- **Error handling**  To allow the LOAD TABLE operation when the statement encounters errors while inserting or parsing rows, problem rows can be written to a ROW LOG and an error message written
to a MESSAGE LOG. After the specifiable number of errors is encountered, the statement fails and rolls back. The following clauses have been added to support this feature:

- ALLOW...ERRORS
- ROW LOG...
- MESSAGE LOG...

See “LOAD TABLE statement” [SQL Anywhere Server - SQL Reference].

Programming interface behavior changes

Following is a list of behavior changes to programming interfaces introduced in SQL Anywhere version 16.0.

- **Out of memory fatal error** On Windows, previous versions of the client libraries could attempt to display an Out of memory window when a client library routine was not able to acquire dynamic memory. It would wait for a user response and then terminate the application. The client libraries no longer do this. Now when an out of memory condition arises, a Windows event log entry is written under event source SQLANY16.0 (32-bit applications) or SQLANY64 16.0 (64-bit applications). After writing the event log entry, the software continues to execute permitting the requesting routine to handle this error condition. This change prevents client applications that are written as Windows services from hanging. It is up to the client application developer to determine the cause of an application exhausting all available memory. See “Formatting Event Log messages” [SQL Anywhere Server - Programming].

- **ODBC SQLTables function** The SQLTables function has been corrected to categorize the dbo-owned system tables installed by the CREATE DATABASE statement as SYSTEM TABLE rather than TABLE (in the TABLE_TYPE column of the result set). Tables such as dbo.jdbc_function_escapes and dbo.spt_jdatatype_info are now classified as system tables.

The SQLTables function has been corrected to categorize the dbo-owned system views installed by the CREATE DATABASE statement as SYSTEM VIEW rather than VIEW (in the TABLE_TYPE column of the result set). Views such as dbo.systypes and dbo.sysusers are now classified as system views.

- **JDBC DatabaseMetaData.getTables method** The getTables method of the DatabaseMetaData class has been corrected to categorize the dbo-owned system tables installed by the CREATE DATABASE statement as SYSTEM TABLE rather than TABLE (in the TABLE_TYPE column of the result set). Tables such as dbo.jdbc_function_escapes and dbo.spt_jdatatype_info are now classified as system tables.

The getTables method has been corrected to categorize the dbo-owned system views installed by the CREATE DATABASE statement as SYSTEM VIEW rather than VIEW (in the TABLE_TYPE column of the result set). Views such as dbo.systypes and dbo.sysusers are now classified as system views.

- **Database Tools C API changes** The script_full_path field has been removed from the a_sync_db structure. Use of this field has not been supported since version 8.x. See “a_sync_db structure [database tools]” [SQL Anywhere Server - Programming].
New fields (wait_before_start, wait_after_end, backup_comment, auto_tune_writers, and backup_history) and enumerations have been added to the a_backup_db structure. See “a_backup_db structure [database tools]” [SQL Anywhere Server - Programming].

- **Identifier rules and datatype limitations now enforced for JSON objects**  
  JSON object identifiers must comply with the identifier rules defined in the database server. As well, the database server enforces the same limits for JSON data types as it does for the underlying ROW and ARRAY data types.

### SQL statement behavior changes

Following is a list of SQL behavior changes introduced in SQL Anywhere version 16.0.

**ALTER DATABASE statement**  
By default, the database is now stopped and restarted after an upgrade. To prevent the database from restarting, use the RESTART OFF clause to stop the database after the upgrade. See “ALTER DATABASE statement” [SQL Anywhere Server - SQL Reference].

**SELECT * supported in view definitions**  
In previous releases, SELECT * was only supported in the main query of the CREATE VIEW statement. Now it is supported in the main query, a subquery, a derived table, or a subselect of the CREATE VIEW statement. See “CREATE VIEW statement” [SQL Anywhere Server - SQL Reference].

**START DATABASE statement**  
The ON clause of the START DATABASE statement is deprecated. The START DATABASE statement can only start databases on the current database server. Previously, you could use the ON clause in Interactive SQL to specify a database server to use. See “START DATABASE statement” [SQL Anywhere Server - SQL Reference].

**ALTER SERVER statement**  
The CONNECTION CLOSE clause of ALTER SERVER statement is not supported. Use the DROP REMOTE CONNECTION statement. See “DROP REMOTE CONNECTION statement” [SQL Anywhere Server - SQL Reference].

**LOAD TABLE statement**

- **STRIP ON clause**  
The STRIP ON clause of the LOAD TABLE statement is not supported. Use the STRIP RTRIM clause instead. See “LOAD TABLE statement” [SQL Anywhere Server - SQL Reference].

**SYSUSER system view now stores standalone role information**  
Previously, the SYSUSER system view stored only users in the system. Now, standalone roles (roles not associated with a user) are also stored in this view as well. However, only the user_id, object_id, user_name, and user_type columns are meaningful for these roles. See “SYSUSER system view” [SQL Anywhere Server - SQL Reference].

### SQL Anywhere deprecated and discontinued features

**Deprecated features**

The following is a list of deprecated features in version 16.0:
• **Authorities deprecated**  SQL Anywhere 16.0 introduces a new role- and privileged-based security model to replace the former authorities and permissions security model. The new role-based security model provides you with granular control over the privileged tasks that users can perform, and simpler administration of access control. The former database authorities are now deprecated and have been replaced by either system roles or compatibility roles:
  ○ DBA authority
  ○ REMOTE DBA authority
  ○ BACKUP authority
  ○ VALIDATE authority
  ○ PROFILE authority
  ○ READCLIENTFILE authority
  ○ WRITECLIENTFILE authority
  ○ READFILE authority
  ○ WRITEFILE authority
  ○ RESOURCE authority

  See “What happened to authorities, permissions, and groups?” [SQL Anywhere Server - Database Administration].

• **java_location and java_main_userid options**  The java_location and java_main_userid options are deprecated since version 11.0.0 with the introduction of the ISYSEXTERNENV system table. Applications that have been using the java_location option to identify which specific Java VM to use should use the ALTER EXTERNAL ENVIRONMENT statement instead to set the location for the Java VM in the ISYSEXTERNENV table. There is no alternative for the java_main_userid option. See “SQL Anywhere external environment support” [SQL Anywhere Server - Programming].

• **secure_feature_key database option**  The secure_feature_key database option is deprecated. Use the sp_use_secure_feature_key system procedure. See “sp_use_secure_feature_key system procedure” [SQL Anywhere Server - SQL Reference].

• **PHP functions**  The following PHP functions are deprecated. Each of these functions has a newer equivalent with a name starting with sasql_ instead of sqlanywhere_.

<table>
<thead>
<tr>
<th>Deprecated PHP functions</th>
<th>New PHP function equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqlanywhere_commit</td>
<td>“sasql_commit” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_connect</td>
<td>“sasql_connect” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_data_seek</td>
<td>“sasql_data_seek” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_disconnect</td>
<td>“sasql_disconnect” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_error</td>
<td>“sasql_error” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_errorcode</td>
<td>“sasql_errorcode” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_execute</td>
<td></td>
</tr>
</tbody>
</table>
### Deprecated PHP functions

<table>
<thead>
<tr>
<th>Deprecated PHP functions</th>
<th>New PHP function equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqlanywhere_fetch_array</td>
<td>“sasql_fetch_array” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_fetch_field</td>
<td>“sasql_fetch_field” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_fetch_object</td>
<td>“sasql_fetch_object” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_fetch_row</td>
<td>“sasql_fetch_row” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_free_result</td>
<td>“sasql_free_result” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_identity</td>
<td></td>
</tr>
<tr>
<td>sqlanywhere_num_fields</td>
<td>“sasql_num_fields” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_num_rows</td>
<td>“sasql_num_rows” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_pconnect</td>
<td>“sasql_pconnect” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_query</td>
<td>“sasql_query” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_result_all</td>
<td>“sasql_result_all” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_rollback</td>
<td>“sasql_rollback” [SQL Anywhere Server - Programming]</td>
</tr>
<tr>
<td>sqlanywhere_set_option</td>
<td>“sasql_set_option” [SQL Anywhere Server - Programming]</td>
</tr>
</tbody>
</table>

### Discontinued features

The following is a list of discontinued features in version 16.0:

- **SQL Anywhere JDBC 3.0 driver**  The SQL Anywhere JDBC 3.0 driver is not included in this software release. Applications using sajdbc.jar must switch to sajdbc4.jar. See “JDBC support” [SQL Anywhere Server - Programming].

- **JDBC-based remote data access server classes**  The JDBC-based remote server classes are not supported in this software release. These classes include the ASEJDBC, IQJDBC, and SAJDBC remote server classes. See “CREATE SERVER statement” [SQL Anywhere Server - SQL Reference].

- **OLE DB provider no longer supports Borland Delphi**  The SQL Anywhere OLE DB provider can no longer be used with old versions of Borland Delphi. Borland Delphi does not support the DBTYPE_DBTIME2 and DBTYPE_DBTIMESTAMPOFFSET data types. Embarcadero no longer supports versions of Delphi from 2006 and earlier. See http://www.embarcadero.com/products/delphi/previous-versions.

- **Address Windowing Extensions (AWE) unsupported**  The use of Address Windowing Extensions for 32-bit Windows is unsupported, and the -cw and -cm database server options are no longer supported. If you need a large cache, use the 64-bit version of the SQL Anywhere database server on a 64-bit operating system.
MobiLink new features

Following is a list of additions to MobiLink introduced in version 16.0. For information about supported operating systems and versions, see http://www.sybase.com/detail?id=1061806.

- **MobiLink Profiler replaces the MobiLink Monitor** The MobiLink Monitor has been removed. Use the MobiLink Profiler to get detailed performance information about synchronizations during development and testing. Use the SQL Anywhere Monitor for production monitoring of a MobiLink deployment. See “MobiLink Profiler” [MobiLink - Server Administration].

- **Java and .NET data scripts can no longer return SQL** Java and .NET synchronization scripts can no longer return strings that are interpreted by the MobiLink server as SQL statements. Java and .NET scripts that return SQL now produce errors and cause the synchronization to fail.

  It is recommended that you use direct row handling for Java or .NET scripts that need to upload data to and download data from the consolidated database. See “Direct row handling” [MobiLink - Server Administration].

- **MobiLink server backwards and forwards compatibility** Any build of a version 16 MobiLink client can synchronize with any build of the version 16 MobiLink server. Previously, the MobiLink server build needed to be greater than or equal to the MobiLink client build.

- **Support for the XMLTYPE data type** Synchronization with columns using the Oracle XMLTYPE data type is now supported. See “Oracle XMLTYPE data type” [MobiLink - Server Administration].

- **MobiLink server supports integrated LDAP authentication** MobiLink user authentication against Lightweight Directory Access Protocol (LDAP) servers is now available in MobiLink server. LDAP authentication is implemented using the following new MobiLink stored procedures:
  - ml_addldap_server system procedure Use this system procedure to add, drop and update LDAP servers. See “ml_add ldap_server system procedure” [MobiLink - Server Administration].
  - ml_add_certificates_file stored procedure Use this system procedure to set up the trusted certificates required for encrypted TLS communication between the MobiLink server and an LDAP server. See “ml_add_certificates_file system procedure” [MobiLink - Server Administration].
  - ml_add_user_auth_policy stored procedure Use this system procedure to set up MobiLink user authentication policies. See “ml_add_user_auth_policy system procedure” [MobiLink - Server Administration].

- **New ml_model_drop system procedure** Use this system procedure to drop a synchronization model and its associated schema. Only synchronization models added using the MobiLink 16.0 plug-in for Sybase Central can be dropped. See “ml_model_drop system procedure” [MobiLink - Server Administration].

- **New ml_model_check_all_schema system procedure** Use this procedure to check the status of each schema object required by deployed synchronization models. This stored procedure returns
information for all deployed synchronization models. See “ml_model_check_all_schema system procedure” [MobiLink - Server Administration].

- **New ml_model_check_version_schema system procedure** Use this procedure to check the status of each schema object required by deployed synchronization models. This stored procedure returns information for the specified script version. See “ml_model_check_all_schema system procedure” [MobiLink - Server Administration].

- **MobiLink server and clients now support TLS version 1.1** MobiLink server and MobiLink clients now support TLS version 1.1. This applies to both direct TLS and HTTPS synchronization. TLS version 1.1 is not supported on Mac OS X.

- **Reporting locking and blocking information** By default, the MobiLink server now reports locking and blocking information for long running user-defined scripts to the MobiLink Profiler, in addition to logging the information to the MobiLink server log file.

  Use the ml_add_property stored procedure to set the properties for this feature. See “ml_add_property system procedure” [MobiLink - Server Administration].

- **-ts option for event tracing** Use the new -ts option with the mlsrv16, dbmlsync and dblsn utilities to set up a tracing session.

  See:

  - “-ts mlsrv16 option” [MobiLink - Server Administration]
  - “-ts dbmlsync option” [MobiLink - Client Administration]
  - “-ts dblsn option” [MobiLink - Server-Initiated Synchronization]

### Consolidated databases

Following is a list of enhancements to consolidated database support for MobiLink introduced in SQL Anywhere version 16.0.


- **SQL Anywhere 16 consolidated database support** The MobiLink server now supports SQL Anywhere 16 consolidated databases. See “SQL Anywhere consolidated database” [MobiLink - Server Administration].

- **SAP HANA 1.0 database server support** The MobiLink server now supports SAP HANA database servers, version 1.00.32 and later. See “SAP HANA consolidated database” [MobiLink - Server Administration].

- **SAP Sybase IQ 15.4 consolidated database support** The MobiLink server now supports consolidated databases running on SAP Sybase IQ 15.4 (ESD #1 and later) servers. See “SAP Sybase IQ consolidated database” [MobiLink - Server Administration].

- **Microsoft SQL Server 2012 consolidated database support** The MobiLink server now supports consolidated databases running on Microsoft SQL Server 2012 servers. This support is
limited to MobiLink servers running on Microsoft Windows only. See “Microsoft SQL Server consolidated database” [MobiLink - Server Administration].

- **MySQL server 5.5.X (X must be 16 or later) consolidated database support**  The MobiLink server now supports consolidated databases running on My SQL servers version 5.5.x, where x is 16 or later. See “MySQL consolidated database” [MobiLink - Server Administration].

- **Sybase Adaptive Server Enterprise 15.7 consolidated database support**  The MobiLink server now supports consolidated databases running on a Sybase ASE 15.7 server. See “Adaptive Server Enterprise consolidated database” [MobiLink - Server Administration].

- **IBM DB2 LUW 10.1 consolidated database support**  The MobiLink server now supports IBM DB2 LUW 10.1 consolidated databases. See “IBM DB2 LUW consolidated database” [MobiLink - Server Administration].

- **Oracle 11g consolidated database support**  The MobiLink server now supports Oracle 11g consolidated databases. See “Oracle consolidated database” [MobiLink - Server Administration].

**MobiLink server**

Following is a list of enhancements to the MobiLink server introduced in SQL Anywhere version 16.0.

**New mlsrv16 features**

- **MobiLink server now supports multi-threaded network processing**  The MobiLink server now supports multiple network worker threads processing its network streams concurrently. Use the new -wn mlsrv16 option to set stream threads. See “-wn mlsrv16 option” [MobiLink - Server Administration].

- **header_limit option for mlsrv16 -x option**  The new header_limit option for the mlsrv16 -x option enables you to specify the maximum amount of header data that is allowed in a given request. It can be used with the HTTP, HTTPS, and OE protocols. See “-x mlsrv16 option” [MobiLink - Server Administration].

- **trusted_certificates option for mlsrv16 -x option**  When encryption is in effect, the trusted_certificates option provides validation of clients. Use this option with the NetworkData.ClientCertificates API so MobiLink can more fully integrate with your enterprise certificate infrastructure.

  See:

  - “-x mlsrv16 option” [MobiLink - Server Administration]
  - “NetworkData interface [MobiLink server Java]” [MobiLink - Server Administration]
  - “NetworkData interface [MobiLink server .NET]” [MobiLink - Server Administration]

**New MobiLink scripting features**

- **New authentication_message named system parameter**  The MobiLink server now supports the new authentication_message named system parameter. This named parameter is available in all the
user authentication events including the authenticate_user, authenticate_user_hashed, and authenticate_parameters scripts.

See:

- “authenticate_user connection event” [MobiLink - Server Administration]
- “authenticate_user_hashed connection event” [MobiLink - Server Administration]
- “authenticate_parameters connection event” [MobiLink - Server Administration]

**New script_version named system parameter**  The MobiLink server now supports the new script_version parameter to specify that the MobiLink server passes the script version string used for the current synchronization to this parameter. This named parameter is available in all MobiLink server synchronization events, except for the begin_connection, begin_connection_autocommit, and end_connection events. See “Named script parameters” [MobiLink - Server Administration].

**MobiLink server utilities**

Following is a list of enhancements to the MobiLink server utilities introduced in SQL Anywhere version 16.0.

- **New -r option for mluser utility**  Use this option with the mluser -u option to reset the synchronization state for the given username and remote ID. See “MobiLink User Authentication utility (mluser)” [MobiLink - Server Administration].

- **New -n option for mluser utility**  Use this option to register a MobiLink user for LDAP user authentication. See “MobiLink User Authentication utility (mluser)” [MobiLink - Server Administration].

- **New MobiLink arbiter stop utility**  Use the new mlarbstop utility to stop the MobiLink arbiter server. See “MobiLink Arbiter Stop utility (mlarbstop)” [MobiLink - Server Administration].

- **New -sv option for mlreplay**  Use the new -sv option for mlreplay to replace script versions in a replay session with user-defined values. See “MobiLink Replay utility (mlreplay)” [MobiLink - Server Administration].

- **New script version field for -sci option for mlreplay**  The new script version field is the last field in the comma separated simulated client information file specified with the -sci option for mlreplay. See “MobiLink Replay utility (mlreplay)” [MobiLink - Server Administration].

- **New liveness timeout settings for mlreplay**  The MobiLink Replay utility (mlreplay) can change the liveness timeout using the stream options specified with the -x mlsrv16 option on the command line. See “MobiLink Replay utility (mlreplay)” [MobiLink - Server Administration].

**MobiLink Profiler**

The MobiLink Profiler provides detailed information about the performance of your synchronizations. In addition to the phase-level timings that were available in the MobiLink Monitor, the MobiLink Profiler enables more detailed performance analysis and bottleneck identification via event-level timings, event
and phase sampling, and blocking information from the consolidated database when the MobiLink server
detects long running SQL statements.

All profiling results are saved in a SQL Anywhere database, which can be loaded back into the MobiLink
Profiler or queried by another database client. The MobiLink Profiler automatically handles creating,
starting, connecting to and stopping the profiling database.

See “MobiLink Profiler” [MobiLink - Server Administration].

**MobiLink Monitor**

The MobiLink Monitor has been removed. Use the MobiLink Profiler to get detailed performance
information about synchronizations during development and testing. Use the SQL Anywhere Monitor for
production monitoring of a MobiLink deployment. See “MobiLink Profiler” [MobiLink - Server
Administration] and “SQL Anywhere Monitor” [SQL Anywhere Server - Database Administration].

**MobiLink clients**

Following is a list of enhancements to MobiLink clients introduced in SQL Anywhere version 16.0.

- **Role-based access control (RBAC) and synchronization**  In the new SQL Anywhere 16.0
  role- and privileged-based security model, a user must be granted the
  SYS_RUN_REPLICATION_ROLE to run synchronization and the
  SYS_REPLICATION_ADMIN_ROLE to perform administrative tasks related to synchronization.
  Each of these roles include various system privileges. You can grant and revoke roles and privileges
to attain more control over the security of your synchronization environment.

  See:
  - “Security considerations with role-based access control and synchronization” [MobiLink - Client
    Administration]
  - “New security model: Role-based access control (RBAC)” on page 7
  - “User security (roles and privileges)” [SQL Anywhere Server - Database Administration]
  - “GRANT ROLE SYS_REPLICATION_ADMIN_ROLE statement [MobiLink] [SQL Remote]”
    [SQL Anywhere Server - SQL Reference]
  - “GRANT ROLE SYS_RUN_REPLICATION_ROLE statement [MobiLink] [SQL Remote]”
    [SQL Anywhere Server - SQL Reference]
  - “REVOKE ROLE SYS_REPLICATION_ADMIN_ROLE statement [MobiLink] [SQL Remote]”
    [SQL Anywhere Server - SQL Reference]
  - “REVOKE ROLE SYS_RUN_REPLICATION_ROLE statement [MobiLink] [SQL Remote]”
    [SQL Anywhere Server - SQL Reference]

- **New trusted_certificate_name MobiLink client network protocol option**  The
  trusted_certificate_name protocol option enables you to specify trusted root certificates that are stored
  in the remote database. This option is only available for SQL Anywhere clients. See
  “trusted_certificate_name” [MobiLink - Client Administration].
● **HTTP performance improvements**  MobiLink clients now issue fewer HTTP requests than in previous versions.

### Dbmlsync C++ API changes

- **New privileges required for the Connect method**  The privileges required for the userid specified in the Connect method have changed. See “DbmlsyncClient.Connect method [Dbmlsync C++]” [MobiLink - Client Administration].

- **DBSC_Event structure has a new ‘val3’ integer field**  The DBSC_Event structure has a new ‘val3’ integer field. The meaning of the value in the val3 field depends on the type of event. See “DBSC_Event structure [Dbmlsync C++]” [MobiLink - Client Administration].

- **DBSC_EVENTTYPE_DOWNLOAD_COMMITTED now has two parameters**  In the DBSC_Event structure for the event, val1 contains the number of rows inserted or updated during the download and val2 contains the number of rows deleted by the download.

- **DBSC_EVENTTYPE_UPLOAD_START event has been added**  The new event DBSC_EVENTTYPE_UPLOAD_START has been added. The event is generated when dbmlsync begins to send the upload to the MobiLink server.

- **DBSC_EVENTTYPE_UPLOAD_SENT event has been added**  The new event DBSC_EVENTTYPE_UPLOAD_SENT has been added. The event is generated when dbmlsync completes sending the upload to the MobiLink server.

- **DBSC_EVENTTYPE_DOWNLOAD_START event has been added**  The new event DBSC_EVENTTYPE_DOWNLOAD_START has been added. The event is generated when dbmlsync begins to process the download.

### Dbmlsync .NET API changes

- **New privileges required for the Connect method**  The privileges required for the userid specified in the Connect method have changed. See “DbmlsyncClient.Connect method [Dbmlsync .NET]” [MobiLink - Client Administration].

- **DBSC_Event structure has a new ‘val3’ integer field**  The DBSC_Event structure has a new ‘val3’ integer field. The meaning of the value in the val3 field depends on the type of event. See “DBSC_Event structure [Dbmlsync .NET]” [MobiLink - Client Administration].

- **DBSC_EVENTTYPE_DOWNLOAD_COMMITTED now has two parameters**  In the DBSC_Event structure for the event, val1 contains the number of rows inserted or updated during the download and val2 contains the number of rows deleted by the download.

- **DBSC_EVENTTYPE_UPLOAD_START event has been added**  The new event DBSC_EVENTTYPE_UPLOAD_START has been added. The event is generated when dbmlsync begins to send the upload to the MobiLink server.

- **DBSC_EVENTTYPE_UPLOAD_SENT event has been added**  The new event DBSC_EVENTTYPE_UPLOAD_SENT has been added. The event is generated when dbmlsync completes sending the upload to the MobiLink server.
DBSC_EVENTTYPE_DOWNLOAD_START event has been added  The new event DBSC_EVENTTYPE_DOWNLOAD_START has been added. The event is generated when dbmlsync begins to process the download.

SQL statements
Following is a list of SQL enhancements for SQL Anywhere clients and remote MobiLink databases introduced in SQL Anywhere version 16.0.

- **ALTER SYNCHRONIZATION SUBSCRIPTION statement changes**  The ALTER SYNCHRONIZATION SUBSCRIPTION statement now accepts NULL in the ADDRESS clause, the TYPE clause and the SET SCRIPT VERSION clause. See “ALTER SYNCHRONIZATION SUBSCRIPTION statement [MobiLink]” [SQL Anywhere Server - SQL Reference].

- **ALTER SYNCHRONIZATION USER statement changes**  The ALTER SYNCHRONIZATION USER statement now accepts NULL in the ADDRESS clause and the TYPE clause. See “ALTER SYNCHRONIZATION USER statement [MobiLink]” [SQL Anywhere Server - SQL Reference].

- **START SYNCHRONIZATION SCHEMA CHANGE statement changes**  The START SYNCHRONIZATION SCHEMA CHANGE statement now accepts NULL in the SET SCRIPT_VERSION clause. See “START SYNCHRONIZATION SCHEMA CHANGE statement [MobiLink]” [SQL Anywhere Server - SQL Reference].

MobiLink plug-in for Sybase Central
Following is a list of new features and behavior changes for the MobiLink plug-in for Sybase Central introduced in version 16.0.

MobiLink plug-in for Sybase Central new features
Following is a list of new features for the MobiLink plug-in for Sybase Central introduced in version 16.0.

- **Test tool for synchronization models**  The new Test tool on the synchronization model Deployment tab allows you to test the deployment of a synchronization model. See “Testing a synchronization model before deployment” [MobiLink - Getting Started].

  For usage examples, see “Tutorial: Introducing MobiLink” [MobiLink - Getting Started].

- **Create Project Wizard has been simplified**  The Create Project Wizard has been simplified to make it easier to create a project. Some of the significant changes to this wizard are as follows:
  - A consolidated database must be associated with the project.
  - The Synchronization Model page has been removed.
  - A synchronization model is automatically created when a project is created.
  - Each project now has one remote database type, instead of having a type for each remote schema name.
• **Simplified Deploy Synchronization Model Wizard**  The Deploy Synchronization Model Wizard is simpler and easier to use. If a synchronization model has been previously deployed, the wizard now provides the option to deploy the synchronization model with the values used in the previous deployment.

• **Simplified Table Mapping Editor**  The Table Mapping Editor has been redesigned. Some of the significant changes are as follows:
  ○ The Column Mappings, Download Type, Download Deletes, Download Subset, Download Delete Subset, Conflict Handling, and Status tabs all appear in the Details pane.
  ○ Columns for Download Type, Download Deletes, Download Subset, Download Delete Subset, Conflict Detection and Conflict Resolution options are no longer shown in the Table Mapping pane by default, and are read-only if displayed. They can be displayed by right clicking the table header.
  ○ The Conflict Handling tab in the Details pane is used for conflict detection and resolution options, and replaces the Conflict Resolution tab.
  ○ Tabs for the Details pane now appear at the top of the pane instead of the bottom.

• **New MobiLink Server Command Line Properties windows**  The new MobiLink Server Command Line Properties windows allow you to save MobiLink server command lines to include in MobiLink projects.

• **Ability to drop synchronization models added using the MobiLink 16.0 plug-in**  Use the new Remove From Consolidated Database context menu item for synchronization models to drop a synchronization model and its associated schema. Only synchronization models added using the MobiLink 16.0 plug-in for Sybase Central can be dropped.

• **Support for multiple synchronization models in the consolidated database**  Deploying a synchronization model now preserves schema which is identical to the previously installed schema. During deployment you are warned before deploying if there are incompatibilities between versions. In addition, redeploying updates to the same script version automatically removes schema no longer needed by the new version.

• **Improved conflict handling for synchronization models**  All processing of conflict detection and resolution is now done inside the upload_update script and fetches are only needed when a conflict occurs, instead of on every row. This results in a significant performance improvement when there are large numbers of updates.

• **Create Synchronization Model Wizard has been simplified**  Previously the wizard had optional pages for setting various synchronization options. Those optional pages have been removed, since their synchronization options can also be set in the Mappings tab and in the Synchronization Model Properties window.
LDAP support has been added The MobiLink 16 plug-in includes the following wizards and property sheets to support LDAP authentication with MobiLink:

- Create LDAP Server Wizard
- Create User Authentication Policy Wizard
- LDAP Server Properties window
- Authentication Policy Properties

MobiLink system objects can be removed from a database A new popup menu item is available to remove MobiLink system setup from a consolidated database. Choosing this item removes all MobiLink system objects from the consolidated database, and removes the database from the MobiLink project.

Automatically refresh dynamic objects and properties For central administration of remote databases, automatic refreshing is supported for Agents and deployed remote tasks.

Remote tasks can now be exported to a file Remote tasks can be exported to a file with a .mlt extension. See “Exporting a remote task” [MobiLink - Server Administration].

Limited support for the Oracle XMLTYPE data type The MobiLink plug-in for Sybase Central provides limited support for the Oracle XMLTYPE. See “Limitations of synchronization models” [MobiLink - Getting Started].

Column mapping warnings When creating a synchronization model or adding a table mapping to a synchronization model, a warning is now given if any fixed-length character columns in the consolidated tables are mapped to variable-length character columns in the remote tables. This check is only done if the consolidated database is Oracle, IBM DB2 LUW, Microsoft SQL Server or Adaptive Server Enterprise.

MobiLink behavior changes
Following is a list of behavior changes to MobiLink introduced in version 16.0. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

MobiLink server changes
Following is a list of behavior changes to MobiLink servers introduced in version 16.0.

- COMMIT always issued for successful end_synchronization scripts The end_synchronization scripts are passed a synchronization_ok parameter to denote the state of the synchronization (0 if failed synchronization, 1 if successful). Before this change, end_synchronization scripts would be rolled back if the synchronization_ok parameter was zero. Now, if all of the end_synchronization scripts execute successfully, a COMMIT is issued regardless of the value of the synchronization_ok parameter.

To achieve the old behavior, wrap an existing end_synchronization script body in an IF statement. For example:
IF {ml s.synchronization_ok} = 1 THEN
  -- existing script body here
END IF;

- **Test MobiLink connectivity with a web browser** When using HTTP or HTTPS, with or without the Relay Server, you can use a web browser to verify MobiLink server is listening for requests. MobiLink server responds with a **404 Not Found** error that also mentions the MobiLink server's major version. See “MobiLink connectivity” [MobiLink - Server Administration].

- **32-bit MobiLink server is no longer supported on Linux** Only the 64-bit version of the MobiLink server is supported on Linux.

- **Windows 32-bit MobiLink server is no longer supported for some consolidated databases** The Windows 32-bit MobiLink server is not supported for the following types of consolidated databases:
  - Sybase Adaptive Server Enterprise
  - SAP HANA
  - SAP Sybase IQ
  - IBM DB2 LUW
  - MySQL

- **Columns name are now always sent during synchronization** In MobiLink, column names are now always sent during synchronization. See “Send column names” on page 14.

---

**MobiLink Notifier and Listener changes**

Following is a list of behavior changes to the MobiLink Notifier and Listener introduced in version 16.0.

- **Limited visibility of notification address, subject, and content to the error handler** When the optional Notifier error handler is invoked, any character in the address, subject or content of a notification that is not one of the recommended characters is substituted with an asterisk. See “error_handler event” [MobiLink - Server-Initiated Synchronization].

- **SMS listening is now disabled by default** The dblsn -ns option has been removed and SMS listening is now disabled by default. This option is still supported by the software but is ignored if it is specified. See “Secure Listener deployment” [MobiLink - Server-Initiated Synchronization].

  Use the -ls option to enable SMS listening. See “-ls dblsn option” [MobiLink - Server-Initiated Synchronization].

- **UDP listening is now disabled by default** The dblsn -nu option has been removed and UDP listening is now disabled by default. This option is still supported by the software but is ignored if it is specified. See “Secure Listener deployment” [MobiLink - Server-Initiated Synchronization].

  Use the -lu option to enable UDP listening. See “-lu dblsn option” [MobiLink - Server-Initiated Synchronization].

- **New -ls dblsn option** Use the dblsn -ls option to enable SMS listening. See “-ls dblsn option” [MobiLink - Server-Initiated Synchronization].
- **New -lu dblsn option**  Use the dblsn -lu option to enable UDP listening. See “-ls dblsn option” [MobiLink - Server-Initiated Synchronization].

### MobiLink client changes

Following is a list of behavior changes to MobiLink clients introduced in version 16.0.

- **Sample certificates have been moved to a different directory**  Sample X.509 certificates used for TLS communication have been moved from the bin32 and bin64 directories in the SQL Anywhere installation directory to the `C:\Documents and Settings\All Users\Documents\SQL Anywhere 16\Samples\Certificates` directory, which is shown as `%SQLANYSAMP16\Certificates` in the documentation.

  For more information, see `C:\Documents and Settings\All Users\Documents\SQL Anywhere 16\Samples\Certificates\readme.txt`.

- **Roles and privileges with hooks and stored procedures**  When working with hooks and stored procedures, the following changes are required as a result of the new roles and privileges security model:
  
  - Users must have the MANAGE REPLICATION system privilege to create a hook. See “Event hooks for SQL Anywhere clients” [MobiLink - Client Administration].
  
  - To ensure access to #hook_dict table, hooks must either be owned by a user with the SELECT ANY TABLE and UPDATE ANY TABLE system privileges or be defined using the SQL SECURITY INVOKER clause of the CREATE PROCEDURE statement.
  
  - To ensure access to #hook_dict table, stored procedures for scripted upload must either be owned by a user with the SELECT ANY TABLE and UPDATE ANY TABLE system privileges or have been created using the SQL SECURITY INVOKER clause of the CREATE PROCEDURE statement. See “Stored procedures for scripted upload” [MobiLink - Client Administration].

### MobiLink plug-in for Sybase Central changes

Following is a list of behavior changes for the MobiLink plug-in for Sybase Central introduced in version 16.0.

- **Notification support in synchronization models has been removed**  Synchronization model support for setting up server-initiated synchronization via the Notifications tab has been removed. Server-initiated synchronization is now supported through remote tasks in synchronization projects. There is no longer a Notifications tab and synchronization model deployment no longer supports server-initiated synchronization. To use server-initiated synchronization via the notifier and listener, you can do so independently of your synchronization model.

- **New remote table owner names no longer have dbo as owner**  When creating a synchronization model with a Microsoft SQL Server or Sybase Adaptive Server Enterprise consolidated database with tables owned by `dbo` and a new SQL Anywhere remote database, the
remote tables are not owned by **dbo**. Instead, the consolidated database user is used for the owner of the new remote tables.

## MobiLink deprecated and discontinued features

**Note**
As with all forward-looking statements, the lists of deprecated features are not guaranteed to be complete and are subject to change.

- **SQL Anywhere 11.0.1 consolidated database support**  
  SQL Anywhere version 11.0.1 and earlier is no longer supported.

- **Sybase Adaptive Server Enterprise 15.0 consolidated database support**  
  Sybase Adaptive Server Enterprise version 15.0 and earlier is no longer supported.

- **Microsoft SQL Server 2005 consolidated database support**  
  Microsoft SQL Server version 2005 and earlier is no longer supported.

- **SAP Sybase IQ 15.2 consolidated database support**  
  SAP Sybase IQ version 15.2 and earlier is no longer supported.

- **IBM DB2 LUW 9.5 consolidated database support**  
  IBM DB2 LUW 9.5 and earlier is no longer supported.

- **mlstop -f option removed**  
  The -f command line option for mlstop has been removed. However, if the -f option is specified, mlstop treats it as the -h option. See “MobiLink Stop utility (mlstop)” [MobiLink - Server Administration].

- **Callbacks removed from mlreplay**  
  The following callbacks have been removed from the MobiLink Replay utility (mlreplay):
  - FreeAllUploadRows
  - GetNumUploadTables
  - GetUploadTable
  - GetNumRows
  - GetRow

- **script_full_path field removed**  
  The script_full_path field has been removed from the a_sync_db structure. See “a_sync_db structure [database tools]” [SQL Anywhere Server - Programming].

- **Conflict detection that relies on modified row counts has been removed**  
  Conflict detection that relies on the MobiLink server assessing how many rows were changed by an upload script has been removed. The MobiLink server supports using an upload_fetch script to detect conflicts, or using no conflict detection. See “Conflict detection” [MobiLink - Server Administration].

- **conflicted_inserts and conflicted_deletes upload statistics have been removed**  
  The conflicted_inserts and conflicted_deletes upload statistics have been removed.
Forced conflict mode has been removed  The previously deprecated forced conflict mode has now been removed from MobiLink.

ml_add_column system procedure has been deprecated  The ml_add_column system procedure has been deprecated. Column names are always sent.

ml_column system table has been deprecated  The ml_column system table has been deprecated. Column names are always sent.

The MobiLink tls_type and e2ee_type protocol options no longer accept ECC as an option  See “Elliptic curve encryption (ECC)” on page 13.

QAnywhere new features

Support for QAnywhere has been deprecated and removed. To administer QAnywhere, you must use the QAnywhere plug-in in an earlier version of Sybase Central. For information about previous versions of QAnywhere, refer to http://dcx.sybase.com/index.html#1201en/sqlanywhere_en12/help_top_index.htm.

Relay Server new features

Following is a list of additions to Relay Server introduced in version 16.0. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

Microsoft IIS 8.0 supported  Relay Server can now be set up with Microsoft IIS 8.0. See “Deploying the Relay Server components to Microsoft IIS 7.0, 7.5, or 8.0” [Relay Server].

Quick setup for Microsoft IIS 7.0, 7.5, and 8.0  An interactive quick setup feature, rs-setupt.bat, is provided as an alternative to the manual procedure. rs-setupt.bat is located in the %SQLANY16%/RelayServer/IIS/quicksetup_iis7or8 directory and performs the following tasks:

1. Installs and configures IIS
2. Creates a demo application to verify that the Relay Server is operational
3. Generates a quick reference guide

See “Deploying the Relay Server components to Microsoft IIS 6.0 on Windows Server 2003” [Relay Server] and “Deploying the Relay Server components to Microsoft IIS 7.0, 7.5, or 8.0” [Relay Server].

Apache 2.4 supported  Apache 2.4 is now supported.

See “Deploying the Relay Server components to Apache on Linux” [Relay Server].

Quick setup for Apache on Linux  The quick setup feature performs the following tasks:

1. Configures the Apache web server for Relay Server. This step can be accomplished by running ap-setupt.sh script in the install-dir/relayserver/quicksetup_apache directory.
2. Creates and starts Relay Server test services This step can be accomplished by running `rs-test-setup.sh` script in the `install-dir/relayserver/quicksetup_apache` directory.

3. Generates a quick reference guide

See “Deploying the Relay Server components to Apache on Linux” [Relay Server].

- **Improved troubleshooting for Outbound Enabler**  
  Outbound Enabler error messages have been improved with error codes and more detailed problem resolution suggestions to assist in troubleshooting.

- **Improved server response flow control**  
  On occasion, shared memory resources can become exhausted due to issues with excessive buffering of server responses, which can include large numbers of clients, slow-reading clients, or large HTTP responses. To improve memory usage, Relay Server provides a new configuration option, `max_client_buffer = memory size`, allowing you to specify a limit on the memory buffer size for each client. The default value is 1 MB. The maximum value is 4 GB. See “Backend farm section” [Relay Server].

- **Renew overlapped cookie**  
  In the case where a client is sharing an overlapping standard cookie across multiple concurrent connections, timeout errors can occur. This issue will appear in the Relay Server log. When `renew_overlapped_cookie` is set on, Relay Server detects the cookie overlap for the farm that has this property explicitly turned on and renews the overlapping cookie by creating a new affinity binding. See “Backend farm section” [Relay Server].

- **Improved proxy support using automatic passthrough**  
  Proxies use HTTPS passthrough using a CONNECT request to establish a non-SSL-terminating tunnel. This new feature allows data streaming on Outbound Enabler connections to the Relay Server through buffering proxies.

- **up_pad_size setting (rs.config options section)**  
  This setting inserts a pad of maximum transmission unit (MTU) size to improve latency when sparse numbers of small requests are being uploaded via the RSOE channel. Latency for the Outbound Enabler is also improved for server responses. See “Options section” [Relay Server].

- **Remote administration of the Relay Server log file**  
  Remote administration of the Relay Server log file is supported through the `AdminChannel` class in `rstool.jar`, providing:
  
  - Remote configuration update and retrieval
  - Remote Relay Server log extraction
  - Local Outbound Enabler log extraction
  - Relay Server log archiving

  See “Remote administration of the Relay Server log file” [Relay Server].

---

**Relay Server behavior changes**

Following is a list of behavior changes to Relay Server introduced in version 16.0. For information about supported platforms and versions, see [http://www.sybase.com/detail?id=1061806](http://www.sybase.com/detail?id=1061806).
● **Configuration option to limit memory consumption**  The shared_mem = memory size option in the Options section of the configuration file now specifies the maximum amount of shared memory that the Relay Server uses for state tracking and growth. See “Options section” [Relay Server].

● **Relay Server status page displays greater detail**  The Relay Server status page has been enhanced to allow status to be displayed for specific farms and backend servers, as well as system-wide status. Additionally you can set the frequency of the status page refresh rate. See “Relay Server status page” [Relay Server].

● **Relay Server logging changes**  Relay Server logging now supports the following:

  ○ The Relay Server and Outbound Enabler logs now report timestamps in millisecond resolution using RFC 822 local differential format (+/- hhmm).

  ○ When the Relay Server and Outbound Enabler handle HTTP requests that carry the SAP Passport header, the Relay Server and the Outbound Enabler increase the request handling verbosity level to match the trace level requirement contained in the Passport and also add suffixes to the associated log lines to list the key information of the Passport. See “Relay Server logging and SAP Passports” [Relay Server].

### Relay Server plug-in for Sybase Central changes

Following is a list of behavior changes for the Relay Server plug-in for Sybase Central introduced in version 16.0.

● **New Network tab for Relay Server Farm Properties**  In the Relay Server Farm Properties window, there is a new Network tab. On this tab, you can enable sending padding packets to reduce network latency.

● **New Advanced tab for Backend Server Farm Properties**  In the Backend Server Farm Properties window, there is a new Advanced tab. On this tab, you can set:

  ○ **Client-Server Affinity**  Formerly Affinity on the General tab.

  ○ **Maximum Client Buffer Size**  Lets you set the new max_client_buffer option.

  ○ **Renew Overlapped Cookie**  In the case where a client device is sharing a standard cookie across multiple concurrent overlapping connections, timeout errors can occur. This issue appears in the Relay Server log. When Renew Overlapped Cookie is set to on, Relay Server detects the cookie overlap for the farm that has this property explicitly turned on and renews the overlapping cookie by creating a new affinity binding.

See also

● “Options section” [Relay Server]
● “Affinity” [Relay Server]
● “Backend farm section” [Relay Server]
Relay Server deprecated and discontinued features

Note
As with all forward-looking statements, the lists of deprecated features are not guaranteed to be complete and are subject to change.

- **32-bit Relay Server for Windows (Microsoft IIS) discontinued**  The 32-bit version of Relay Server for Windows is discontinued and is no longer installed.

- **32-bit Relay Server for Apache on Linux discontinued**  The 32-bit version of Relay Server for Windows is discontinued and is no longer installed.

- **Auto start no longer supported**  The start (auto start) option is no longer supported as a property in the Options section of the Relay Server configuration file.

- **ECC**  The Outbound Enabler tls_type option no longer supports ECC as a choice. The tls_type protocol option has been removed from the documentation.

UltraLite new features

Following is a list of additions to UltraLite introduced in version 16.0. For information about supported platforms and versions, see [http://www.sybase.com/detail?id=1061806](http://www.sybase.com/detail?id=1061806).

General features

- **New download synchronization statistics are available**  UltraLite now supports three new synchronization statistics for downloads in the syssyncresult system table—ignored_updates, ignored_deletes, and truncate_deletes.

  The ignored_updates parameter indicates the number of duplicate rows that are received in the download. The ignored_deletes parameter indicates the number of deleted rows received in the download of rows that have already been deleted. The truncate_deletes parameter indicates the number of rows that were deleted in the download by a truncate operation, which is accomplished by returning NULL column values in the download_delete_cursor script.

  See:

  - “syssyncresult system table” [UltraLite - Database Management and Reference]
  - “ul_sync_stats_download structure [UltraLite C and Embedded SQL datatypes]” [UltraLite - C and C++ Programming]
  - “ul_sync_stats_upload structure [UltraLite C and Embedded SQL datatypes]” [UltraLite - C and C++ Programming]
  - “download_delete_cursor scripts” [MobiLink - Server Administration]
  - “ULSyncProgressData class [UltraLite.NET]” [UltraLite - .NET Programming]

- **ZLIB compression supported on BlackBerry smartphones**  The UltraLiteJ API now supports ZLIB data compression for BlackBerry smartphones when synchronizing with a MobiLink server.
End-to-end (E2EE) encryption supported on BlackBerry smartphones  The UltraLiteJ API now supports end-to-end encryption for BlackBerry smartphones when synchronizing with a MobiLink server.

HTTP authentication supported on BlackBerry smartphones  The UltraLiteJ API now supports HTTP authentication for BlackBerry smartphones.

Custom HTTP headers supported on BlackBerry smartphones  The UltraLiteJ API now supports custom HTTP headers for BlackBerry smartphones.
See:
- “StreamHTTPParms.addCustomHTTPHeader method [BlackBerry] [UltraLiteJ]” [UltraLite - Java Programming]
- “StreamHTTPParms.getCustomHTTPHeaders method [BlackBerry] [UltraLiteJ]” [UltraLite - Java Programming]

● **Index hashing supported on BlackBerry smartphones** Use the WITH MAX HASH SIZE clause to adjust index hash sizes in an UltraLite Java edition database.

See:
- “Optimal hash size limit” [UltraLite - Database Management and Reference]
- “Connection.OPTION_MAX_HASH_SIZE variable [UltraLiteJ]” [UltraLite - Java Programming]
- “CREATE TABLE statement [UltraLite]” [UltraLite - Database Management and Reference]
- “ALTER TABLE statement” [SQL Anywhere Server - SQL Reference]
- “CREATE INDEX statement [UltraLite]” [UltraLite - Database Management and Reference]
- “ALTER INDEX statement” [SQL Anywhere Server - SQL Reference]
- “sysindex system table” [UltraLite - Database Management and Reference]

● **Indexing performance improvements for BlackBerry smartphones** Performance improvements have been made to indexing to reduce the number of pages written when committing inserts, updates, and deletes.

Fewer page reads and writes are needed when single inserts are performed. CPU work is reduced when a block of inserts is performed.

Sorting your download and download delete cursor synchronization server scripts by primary key is recommended when synchronizing with BlackBerry smartphones. See “Scripts to download rows” [MobiLink - Server Administration].

● **Error reporting improvements for UltraLite BlackBerry smartphones when multiple applications connect to a database concurrently** A SQLE_FILE_IN_USE error is now reported if more than one application attempts to connect to an UltraLite Java edition database concurrently. See “UltraLite and UltraLite Java edition database limitations” [UltraLite - Database Management and Reference].

● **AES 256-bit encryption and obfuscation for BlackBerry smartphones is now built in** You can enable encryption or obfuscation in your BlackBerry application by using the ConfigPersistent.enableAesDBEncryption or ConfigPersistent.enableObfuscation method in the UltraLiteJ API.

An -ek option has been added to several UltraLite Java edition utilities so that you can specify the encryption key needed to access encrypted databases.
Custom user authentication messages can now be retrieved

Custom user authentication synchronization scripts on the MobiLink server now provide additional details with respect to the results of a synchronization's authentication.

A new system table, syssyncresult, has been added to the UltraLite management system to support this feature. This system table is a representation of the ul_sync_result structure.

For UltraLite Java edition, the SyncResult.getAuthMessage method can be used to retrieve the custom user authentication information.

Platforms and devices

RSA TLS and HTTPS synchronization is supported for Pocket PC 2003

Pocket PC 2003 now supports RSA TLS and HTTPS synchronization. RSA FIPS 140-2 synchronization is not supported.
Programming interfaces

UltraLite C/C++

- **Upload and download statistics of synchronization streams are reported separately**
  The `ul_sync_stats` structure has been split into two separate structures to improve statistic reports: `ul_sync_stats_download` and `ul_sync_stats_upload`.

  See:
  
  - “`ul_sync_stats_download` structure [UltraLite C and Embedded SQL datatypes]” [UltraLite - C and C++ Programming]
  - “`ul_sync_stats_upload` structure [UltraLite C and Embedded SQL datatypes]” [UltraLite - C and C++ Programming]

- **Wide character (UTF-16/UCS-2) data streaming support for iOS**
  Wide character support for iOS has been added to character data streaming methods.

  See:
  
  - “ULResultSet.GetStringChunk method [UltraLite C++]” [UltraLite - C and C++ Programming]
  - “ULResultSet.AppendStringChunk method [UltraLite C++]” [UltraLite - C and C++ Programming]

- **Applications can monitor download progress**
  Two new fields, current_download_row_count and total_download_row_count, have been added to the `ul_sync_status` structure to allow applications to monitor the progress of a download.

  See:
  
  - “Synchronization status information” [UltraLite - C and C++ Programming]
  - “`ul_sync_status` structure [UltraLite C and Embedded SQL datatypes]” [UltraLite - C and C++ Programming]

UltraLite.NET

- **Applications can monitor download progress**
  Two new properties, `CurrentDownloadRowCount` and `TotalDownloadRowCount`, have been added to the `ULSyncProgressData` class to allow applications to monitor the progress of a download.

  See:
  
  - “`ULSyncProgressData.TotalDownloadRowCount` property [UltraLite.NET]” [UltraLite - .NET Programming]
UltraLiteJ

- **Reporting of row statistics during synchronization has improved** Additional details, which involve the number and types of operations that have been sent and received during a synchronization, are reported.

Once the synchronization enters the SyncObserver.State.RECEIVING_TABLE state for the first table, the SyncResult.getTotalDownloadRowCount method returns the total number of rows expected in the download.

The getSentRowCount method has been replaced with the getSentDeletes, getSentInserts and getSentUpdates methods.

The getReceivedRowCount method has been augmented with the getReceivedDeletes, getReceivedIgnoredDeletes, getReceivedIgnoredUpdates, getReceivedInserts, getReceivedTruncateDeletes, and getReceivedUpdates methods. These new methods indicate how the download has been applied to the local UltraLite Java edition database. The getReceivedRowCount method gives the current row count during the download.

The getReceivedRowCount method now returns long values.

See:
- “SyncResult.getTotalDownloadRowCount method [UltraLiteJ]” [UltraLite - Java Programming]
- “SyncResult.getSentDeletes method [UltraLiteJ]” [UltraLite - Java Programming]
- “SyncResult.getSentInserts method [UltraLiteJ]” [UltraLite - Java Programming]
- “SyncResult.getSentUpdates method [UltraLiteJ]” [UltraLite - Java Programming]
- “SyncResult.getReceivedDeletes method [UltraLiteJ]” [UltraLite - Java Programming]
- “SyncResult.getReceivedIgnoredDeletes method [UltraLiteJ]” [UltraLite - Java Programming]
- “SyncResult.getReceivedIgnoredUpdates method [UltraLiteJ]” [UltraLite - Java Programming]
- “SyncResult.getReceivedInserts method [UltraLiteJ]” [UltraLite - Java Programming]
- “SyncResult.getReceivedTruncateDeletes method [UltraLiteJ]” [UltraLite - Java Programming]
- “SyncResult.getReceivedUpdates method [UltraLiteJ]” [UltraLite - Java Programming]
- “SyncResult.getReceivedRowCount method [UltraLiteJ]” [UltraLite - Java Programming]

- **New methods can be used to manage the input parameters of a prepared statement on Android smartphones** The PreparedStatement.getParameterCount and PreparedStatement.getParameterType methods have been added to count the number of parameters in a prepared statement and to get the domain type of a parameter, respectively.

See:
- “PreparedStatement.getParameterCount method [Android] [UltraLiteJ]” [UltraLite - Java Programming]
- “PreparedStatement.getParameterType method [Android] [UltraLiteJ]” [UltraLite - Java Programming]

- **New methods can be used to manage error parameters on Android smartphones** The getParameterCount and getParameter method have been added to count the number of error parameters and to return a specific error parameter, respectively.
A new method can be used to count the number of rows required for upload on Android smartphones

The DatabaseInfo.getNumberRowsToUpload(String, int) method has been added to provide the same functionality as the ULConnection::CountUploadRows(const char *pubList, ul_u_long threshold) method for UltraLite C/C++. See “DatabaseInfo.getNumberRowsToUpload method [UltraLiteJ]” [UltraLite - Java Programming].

The UltraLite database connection state can be reported on Android smartphones

The Connection.getState method is now supported on Android smartphones. See “Connection.getState method [UltraLiteJ]” [UltraLite - Java Programming].

Warning messages can be retrieved on Android smartphones

Use the Connection.getLastWarning method to retrieve a SQLInfo object, which can obtain the SQL code and message of any warning that occurs during the execution of the last SQL statement.

Database validation is supported on Android smartphones

Use the Connection.validateDatabase method to validate an UltraLite database on an Android smartphone. The new ValidateDatabaseProgressData and ValidateDatabaseProgressListener interfaces allow Android applications to track the progress of the validation operation.

Encryption keys can be changed on Android smartphones

Use the Connection.changeEncryptionKey method to change the encryption key of an UltraLite database on an Android smartphone.

• **Restartable downloads are supported on Android smartphones**  
   UltraLite on Android smartphones now has the ability to resume partial downloads that fail due to communication errors or when you abort through the SyncObserver object.

   See:
   - “Connection.rollbackPartialDownload method [Android] [UltraLiteJ]” [UltraLite - Java Programming]
   - “SyncParms.setAdditionalParms method [Android] [UltraLiteJ]” [UltraLite - Java Programming]
   - “SyncParms.getAdditionalParms method [Android] [UltraLiteJ]” [UltraLite - Java Programming]
   - “SyncParms.setKeepPartialDownload method [Android] [UltraLiteJ]” [UltraLite - Java Programming]
   - “SyncParms.getKeepPartialDownload method [Android] [UltraLiteJ]” [UltraLite - Java Programming]
   - “SyncParms.setResumePartialDownload method [Android] [UltraLiteJ]” [UltraLite - Java Programming]
   - “SyncParms.getResumePartialDownload method [Android] [UltraLiteJ]” [UltraLite - Java Programming]
   - “SyncResult.getPartialDownloadRetained method [Android] [UltraLiteJ]” [UltraLite - Java Programming]

• **UltraLite events are supported on Android smartphones**  
   UltraLite on Android smartphones now supports events for table modifications, commits, and synchronization completion. The ULjEvent interface has been added to support this feature.

   An Android application registers and waits for events on a Connection object. A Connection can receive notifications of events that are triggered from Connections on other threads.

   See:
   - “ULjEvent interface [Android] [UltraLiteJ]” [UltraLite - Java Programming]
   - “Connection.registerForEvent method [Android] [UltraLiteJ]” [UltraLite - Java Programming]
   - “Connection.unregisterForEvent method [Android] [UltraLiteJ]” [UltraLite - Java Programming]
   - “Connection.waitForEvent method [Android] [UltraLiteJ]” [UltraLite - Java Programming]
   - “Connection.cancelWaitForEvent method [Android] [UltraLiteJ]” [UltraLite - Java Programming]

• **Lazy load indexing options have improved on BlackBerry smartphones**  
   In previous versions, turning off lazy load indexes would have no effect when the row score flush size is greater than zero. Lazy loading would be turned off only when lazy load indexes are turned off and the row score flush size is less than or equal to zero.

   Using the setRowScoreFlushSize method to set the row score flush to a value greater than zero now marks lazy loading as turned on. Setting the setLazyLoadIndexes method to false now sets the row score flush size to zero.
UltraLite behavior changes and deprecated features

Following is a list of deprecated features and behavior changes to UltraLite introduced in version 16.0. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

**Note**
As with all forward-looking statements, the lists of deprecated features are not guaranteed to be complete and are subject to change.

**Behavior changes**

- **ECC support has been removed**  
  UltraLite C/C++ and UltraLiteJ API methods and properties related to ECC configuration have been removed. See “Elliptic curve encryption (ECC)” on page 13.

- **Visual Studio 2005 project files for the CustDB sample have been removed**  
The Visual Studio 2005 project files for the CustDB sample have been removed from the software.  

  The Visual Studio 2008 project files are now located in the %SQLANYSAMP16%\UltraLite\CustDB directory.  

  You can upgrade the Visual Studio 2008 project files of the CustDB sample by opening them in Visual Studio 2010.

- **Encryption control for BlackBerry smartphones is now built-in**  
The ConfigPersistent.setEncryption method has been removed. Encryption control, where custom encryption code had to be supplied to the UltraLiteJ API to encrypt an UltraLite Java edition database, is no longer supported.

- **UltraLite Java edition databases are automatically checkpointed when statements are committed**  
The following members have been removed from the UltraLiteJ API:

  - ConfigPersistent.setAutoCheckpoint
  - ConfigPersistent.getAutoCheckpoint
  - Connection.checkpoint

- **Support for non-persistent indexes and non-shadow paging stores has been removed for BlackBerry smartphones**  
  UltraLite Java edition databases can no longer be created with non-persistent indexes or shadow paging turned off.
The following members have been removed from the UltraLiteJ API:

- ConfigPersistent.hasPersistentIndexes
- ConfigPersistent.setIndexPersistence
- ConfigPersistent.setShadowPaging

- **Support for write-at-end persistent stores has been removed for BlackBerry smartphones** UltraLite Java edition databases can no longer be created as write-at-end databases.

The following members have been removed from the UltraLiteJ API:

- ConfigPersistent.setWriteAtEnd
- ConfigPersistent.writeAtEnd

- **UltraLiteJ getEncryptionKey and setEncryptionKey methods are synonymous with getDatabaseKey and setDatabaseKey methods, respectively** New ConfigPersistent.getEncryptionKey and ConfigPersistent.setEncryptionKey methods have been added to the UltraLiteJ API.

See:

- “ConfigPersistent.getEncryptionKey method [UltraLiteJ]” [UltraLite - Java Programming]
- “ConfigPersistent.setEncryptionKey method [UltraLiteJ]” [UltraLite - Java Programming]

- **UltraLite Java edition utilities have been renamed** UltraLite Java edition utility names now only contain lowercase characters so that they are consistent with the UltraLite utility names. See “UltraLite Java edition utilities” [UltraLite - Database Management and Reference].

- **Column names are now always sent during synchronization** You can no longer control the sending of column names. See “Send column names” on page 14.

**Removed platforms**

J2ME smartphones are no longer supported by the UltraLiteJ API.

**Removed APIs**

- **UltraLite for M-Business Anywhere has been removed** The UltraLite for M-Business Anywhere API is no longer supported.

- **Previous implementation of UltraLite C/C++ has been removed** The old UltraLite C/C++ API implementation, previously defined in the ulcpp11.cpp and uliface.h files, is no longer supported.

**Administration tools new features**

Following is a list of enhancements to administration tools introduced in version 16.0. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.
Based on JRE 7
The SQL Anywhere administration tools, such as Interactive SQL, Sybase Central, the SQL Anywhere Console utility (dbconsole), and the Monitor, now use and require JRE 7 (Java Runtime Environment 1.7.0).

64-bit support for Linux/Solaris
SQL Anywhere now includes 64-bit administration tools for 64-bit Linux and Solaris. Note that 32-bit versions of the administration tools are still available for 32-bit Linux and Solaris platforms as well as the 64-bit version of the administration tools for Mac OS X.

Connect window changes
Test connection tool for the Connect window
The Test Connection tool in the Connect window can now test whether the information provided for generic ODBC and UltraLite databases results in a proper connection.

Regularly submit performance data to the software development team
To help improve the product, the software has the ability to automatically report performance data to the software development team. Previously, performance data was only sent when a crash occurred. During installation, the option to submit performance data to the software development team is enabled by default. You can also enable this feature:

- In Interactive SQL, by clicking Tools » Options » Support
- In Sybase Central, by clicking Help » SQL Anywhere 16 » Configure Update Checker
- With the Support utility (dbsupport), by specifying the -cc autosubmit option.

OEM users
By default, when you deploy the administration tools, the feature that automatically submits performance data to the software development team is disabled. But, your users can enable or disable this feature within the administration tools. To prevent your users from changing whether their data is submitted or not, set the showPerformanceDataUI option to false in the OEM.ini file.

To enable your users to submit performance data to SAP, you must use the Support utility (dbsupport) with the -cc autosubmit option.

See “Regularly submit performance data” [SQL Anywhere Server - Database Administration].

Sybase Central new features
Following is a list of additions to Sybase Central introduced in version 16.0.

Sybase Central plug-in new features
Following is a list of additions to Sybase Central plug-ins introduced in version 16.0.
SQL Anywhere plug-in new features

**Import and export the layout of entity relationship (ER) diagrams** You can import and export the layout of ER diagrams from Sybase Central. See “Viewing entity-relationship (ER) diagrams” [SQL Anywhere Server - Database Administration].

**Assign login policies to multiple users** You can assign a login policy to multiple users by selecting the users in the Users pane and then clicking File » Set Login Policy. You can also view all the users assigned a specific login policy by opening the Properties window for the login policy and clicking the Users tab.

**Support for text indexes on materialized views.** You can create and administer text indexes on materialized views with the SQL Anywhere plug-in.

**LDAP servers** You can administer LDAP servers with the SQL Anywhere plug-in.

**Support for roles and system privileges** You can administer roles and privileges with the SQL Anywhere plug-in.

**Store certificates in the database** You can add and administer certificates with the SQL Anywhere plug-in.

MobiLink plug-in new features

For a summary of MobiLink plug-in features, see “MobiLink plug-in for Sybase Central”.

Sybase Central behavior changes and deprecated features

Following is a list of changes to Sybase Central introduced in version 16.0.

- **Sybase Central version changes to 16.0** SQL Anywhere 16 includes version 16.0 of Sybase Central.

- **jh.jar and binXX/scvw[LL]1600.jar files are no longer required for deployment of Sybase Central** Sybase Central no longer provides JavaHelp so the following files are not included in the install and are not needed to deploy Sybase Central: jh.jar and binXX/scvw[LL]1600.jar.

SQL Anywhere plug-in behavior changes

- **Automatic prompting for secure_feature_key when a secure feature exception occurs** When you execute SQL statements that use a secure feature, you are prompted to specify the secure feature key before the SQL is executed. The server must be started with a secure feature key (-sk) to make use of this feature.

- **View the utility commands created by the Create Database Wizard** The Create Database Wizard includes a new page at the end of the wizard. This page displays the dbinit utility command that is equivalent to the statement that is executed when you click Finish. Clicking Finish runs the utility command.
Alternatively, you can copy the command to the clipboard, click Cancel to exit the wizard, and then run the command at a command prompt. See “Viewing SQL statements and utility commands generated by wizards” [SQL Anywhere Server - Database Administration].

MobiLink plug-in behavior changes

For a summary of changes to the MobiLink plug-in, see “MobiLink plug-in for Sybase Central” on page 48.

Interactive SQL new features

Following is a list of additions to Interactive SQL introduced in version 16.0.

Support for SAP HANA databases  
Use the -hana command line option for the Interactive SQL utility (dbisql) to connect to an SAP HANA database. Alternatively, in the Connect window in Interactive SQL, click Change Database Type, and then click SAP HANA.

Text completion in the SQL editor, table and column name browsing, procedure name and source browsing, importing and exporting are all supported.

The following Interactive SQL statements are not supported: CONNECT statement, SET CONNECTION statement, and START SERVER statement.

The plan viewer for SAP HANA is new, but works similarly to the other plan viewers. The Spatial Viewer is not supported.

OEMs can disable the Interactive SQL fast launcher  
You can disable the fast launcher feature for Interactive SQL and prevent users from re-enabling it using the OEM.ini file. See “Administration tools configuration” [SQL Anywhere Server - Programming].

OEMs can specify how results are to be displayed in Interactive SQL  
In the OEM.ini file, use the following options to specify how results are to be displayed: defaultShowResultsForAllStatements and defaultShowMultipleResultSets. See “Administration tools configuration” [SQL Anywhere Server - Programming].

Numeric values are displayed right-aligned in textual result sets  
Result sets are displayed as text when run as a console application, or as a windowed application when the Show results as text option is selected. Previously, all values were displayed left-aligned. Now, numeric values are displayed right-aligned. This makes it easier to compare the values.

The error window displays line numbers  
If executing a statement causes an error, the error window that appears now displays the line number of the error.

The text completion feature shows parameter lists  
The Show tool tips option in the Options window in Interactive SQL now displays parameter lists as you type for procedures, functions, and spatial methods. Close the parameter list by pressing Esc.

The text completion feature automatically inserts closing brackets, apostrophes and quotation marks  
The text completion feature inserts the closing punctuation mark when you type an
opening parenthesis, square bracket, brace, apostrophe, or quotation mark. To control the insertion of the closing punctuation marks, click Tools » Options » Editor » Text Completion.

The Import Wizard can read column names from imported text files The Import Wizard can read a list of column names from the first line of the file. This option is available only when importing delimited text files.

Interactive SQL behavior changes and deprecated features

Following is a list of changes to Interactive SQL introduced in version 16.0.

ON clause deprecated from START DATABASE statement

The START DATABASE statement can only start databases on the current server. The ON clause of the START Database statements is deprecated. Previously, you could use this clause in Interactive SQL to specify a database server to use. See “START DATABASE statement” [SQL Anywhere Server - SQL Reference].

SQL Anywhere Monitor new features

Following is a list of new features for the SQL Anywhere Monitor introduced in version 16.0.

- Based on JRE 7 The SQL Anywhere Monitor now uses and requires JRE 7 (Java Runtime Environment 1.7.0).

- Alert Notification Settings Administrators can now associate users with resources using the Email Alert Notification Settings option in the Administration window.

SQL Anywhere Monitor behavior changes

Following is a list of changes to the SQL Anywhere Monitor introduced in version 16.0.

Changes to how metrics are deleted from the Monitor

Metrics must be deleted regularly from the Monitor to limit the amount of disk space used. By default, metrics older than 5 days are deleted daily as part of the maintenance plan.

Previously, you could keep metrics for an indefinite period of time and you could control how much space the metrics occupied. Now, you the longest amount of time that you can keep metrics is 2 weeks, and you cannot specify how much space the metrics occupy. Also, metric data cannot be migrated to a new version of the SQL Anywhere Monitor. The -c option for the Migrator utility has been removed. See “Backing up the Monitor” [SQL Anywhere Server - Database Administration].

Change to the database user ID and password credentials

When adding a database resource to the Monitor, you are prompted to supply the user ID and password credentials for the database. When adding a version 16 database, you must supply the credentials for a user that has the SYS_SAMONITOR_ADMIN_ROLE role. Similarly, when repairing a version 16 database resource, you must supply the credentials for a database user with the same roles.
When prompted to supply the user ID and password credentials to close a connection to a database, you must supply the credentials for a user that has the DROP CONNECTION system privilege.

For resources of pre-16.0 databases, you can continue to supply the credentials for a database user with DBA authority. See “Adding a database resource” [SQL Anywhere Server - Database Administration].

SQL Anywhere Console new features

Following is a list of new features introduced in version 16.0 for the SQL Anywhere Console.

- **Based on JRE 7**  The SQL Anywhere Console utility (dbconsole) now uses and requires JRE 7 (Java Runtime Environment 1.7.0).

- **Enable request logging options**  In the Console tab on the Options window, the Enable request logging option now supports the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocks</td>
<td>Enables logging of details showing when a connection is blocked and unblocked on another connection.</td>
</tr>
<tr>
<td>Host variables</td>
<td>Enables logging of host variable values. The information listed for SQL is also logged.</td>
</tr>
<tr>
<td>Plans</td>
<td>Enables logging of execution plans (short form). Execution plans for procedures are also recorded if Procedures is enabled.</td>
</tr>
<tr>
<td>Procedures</td>
<td>Enables logging of statements executed from within procedures and user-defined functions.</td>
</tr>
<tr>
<td>SQL</td>
<td>Enables logging of START DATABASE, STOP DATABASE, STOP SERVER, EXECUTE IMMEDIATE, COMMIT, ROLLBACK, and DROP STATEMENT statements, as well as statement preparation and execution, option settings, PREPARE TO COMMIT operations, connects and disconnects, beginnings of transactions, statements, cursor explanations, errors, and cursor open, close, and resume.</td>
</tr>
<tr>
<td>Triggers</td>
<td>Enables logging of statements executed from within triggers.</td>
</tr>
<tr>
<td>Other</td>
<td>Enables logging of additional request types not included by SQL, such as FETCH and PREFETCH. Specifying Other automatically enables SQL. Specifying Other can cause the request log file to grow rapidly and could negatively affect server performance.</td>
</tr>
<tr>
<td>Replace log file when logging starts</td>
<td>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.</td>
</tr>
</tbody>
</table>
SQL Remote new features

Following is a list of additions to SQL Remote introduced in version 16.0. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

SQL Remote behavior changes and deprecated features

[ This topic has been updated for build 1823. ]

Note
As with all forward-looking statements, the lists of deprecated features are not guaranteed to be complete and are subject to change.

Following is a list of changes to SQL Remote introduced in version 16.0.

- dbremote offline transaction log retrieval has changed  
  The dbremote utility can now retrieve offline transaction logs from the SQL Anywhere database server instead of accessing them directly. If offline transaction logs are required but the given offline transaction log directory cannot be opened or it does not contain offline transaction log files, then dbremote retrieves the offline transaction logs through the database server. The following restrictions apply:

  ○ The user ID that is used by dbremote to connect to the synchronization database must have the READ FILE and WRITE FILE privileges and all the offline transaction log files must be in the online transaction log directory.

  ○ The database server must have Support Package build number 1823 or later to support this feature.

There is a slight performance penalty when using this feature because the database server must do extra work to retrieve the transaction logs.
What's new in version 12.0.1

For information about new features and behavior changes in versions of SQL Anywhere before version 10, see http://dcx.sybase.com/html/dbwnen10/dbwnen10.html.

12.0.1 addendum: New features and behavior changes

SQL Anywhere 12.0.1 documentation refreshed: This release of the SQL Anywhere 12.0.1 documentation is a more recent version than the documentation provided when SQL Anywhere 12.0.1 was first released February 15, 2011. It includes the new features and behavior changes up to, and including, EBF 3554 (Windows). In particular, it includes changes required to support the new cloud capabilities in the SQL Anywhere OnDemand Edition.

Support added for SQL Anywhere OnDemand Edition

SQL Anywhere OnDemand Edition is a data management solution that enables independent software vendors (ISVs) to take business applications to the cloud and offer software as a service (SaaS). ISVs can build, deploy, and manage large cloud applications.

Some enhancements and behavior changes have been made to SQL Anywhere to support SQL Anywhere OnDemand Edition. Below is a list of the major changes that have occurred. Other changes are described in more depth in the SQL Anywhere OnDemand Edition documentation at http://dcx.sybase.com/cloud100.

- **New login mode value for databases in the cloud**  
  A new value, CloudAdmin, has been added for the login_mode database option. This login mode is for internal use by the cloud. See “login_mode option” [SQL Anywhere Server - Database Administration].

- **Connect window enhancement in Sybase Central and Interactive SQL**  
  The Connect window in Sybase Central and Interactive SQL now provides a Connect to a running database in a cloud option for connecting to a tenant database in the cloud.

- **ODBC Configuration for SQL Anywhere window enhancement**  
  The ODBC Configuration for SQL Anywhere window in the ODBC Data Source Administrator now provides a Connect to a running database in a cloud option for connecting to a tenant database in the cloud.

- **NodeType (NODE) connection parameter enhancement**  
  The NodeType connection value now supports MIRROR and READONLY values:

  **MIRROR**  
  Redirects the application to the tenant's mirror database.

  **READONLY**  
  Redirects the application to any read-only database copy: either a copy node or the mirror copy. If there are no read-only copy nodes, READONLY is equivalent to the MIRROR setting. If there are only read-only copy nodes, READONLY is equivalent to the COPY setting.

  See “NodeType (NODE) connection parameter” [SQL Anywhere Server - Database Administration].
• **New NodeType (NODE) connection parameter behavior change** When connecting to a database in the cloud, the default value for the NodeType connection parameter is PRIMARY and automatically redirects you to the tenant’s primary database. When NodeType is set to DIRECT, no redirection is used and the connection succeeds only if the database is running on the host specified.

See “NodeType (NODE) connection parameter” [SQL Anywhere Server - Database Administration].

• **DatabaseName connection parameter required when connecting to cloud databases** When connecting to a database in the cloud, you must specify the DatabaseName connection parameter. There is no default database name for databases in the cloud. See “DatabaseName (DBN) connection parameter” [SQL Anywhere Server - Database Administration].

• **ServerName connection parameter restriction in the cloud** When connecting to a database in the cloud, you can only use the ServerName connection parameter when the NodeType connection parameter is set to DIRECT. In most cases, you should not specify ServerName or NodeType or DIRECT when connecting to a cloud server. See “ServerName (Server) connection parameter” [SQL Anywhere Server - Database Administration].

• **Enhancements to SQL statements** Enhancements have been made to the following statements to facilitate their use with tenant databases in the cloud:

  ○ “ALTER SYNCHRONIZATION SUBSCRIPTION statement [MobiLink]” [SQL Anywhere Server - SQL Reference]
  ○ “ALTER SYNCHRONIZATION USER statement [MobiLink]” [SQL Anywhere Server - SQL Reference]
  ○ “START SYNCHRONIZATION SCHEMA CHANGE statement [MobiLink]” [SQL Anywhere Server - SQL Reference]

When a database is running in the cloud, some SQL Anywhere statements are not applicable or have restrictions. This information is recorded in the description of the individual functions and system procedures.

• **Limitations on the results from system procedures and functions when run on a database in the cloud** When a database is running in the cloud, some of the SQL Anywhere functions and system procedures only return information about the current tenant database. They do not show information about other tenant databases running on the same cloud server. This information is recorded in the description of the individual functions and system procedures.

• **Some database options, server options, and connection parameters not available for use in the cloud** Some database options, server, options, and connection parameters are not available for use in the cloud. More information can be found in the description for each option or parameter.

**Other SQL Anywhere new features and behavior changes**

• **Specifying IPv6 addresses** For IPv6 addresses that include a port number, you must enclose the address in either square brackets or parentheses. See “IPv6 support in SQL Anywhere” [SQL Anywhere Server - Database Administration].

• **Enhancements to loading tables** In previous releases, you could not perform load table operations on temporary tables in a read-only database. This restriction has been removed.
Previously, the LOAD TABLE default for a database running on a primary, a mirror, or a root node server was WITH FILE NAME LOGGING. Now the LOAD TABLE logging default for these types of databases is WITH ROW LOGGING.

- **Support added for comparing database schemas and making them the same** You can use Sybase Central to compare two databases. The comparison generates SQL statements that you can review to determine the differences between two databases. You can execute the SQL statements to make the one database the same as the other database. See “Comparing database schemas” [SQL Anywhere Server - Database Administration] and “Converting a database schema to match another” [SQL Anywhere Server - Database Administration].

- **Account lockout for DBA users** To ensure that DBA users are not locked out of the database indefinitely if they enter an incorrect password, their accounts are locked for one minute once they exceed the specified number of failed connection attempts defined in their login policy.

- **New directory option for OEM.ini file** The directory option of the [preferences] section in the OEM.ini file specifies the directory used by the administration tools to save user-specific configuration files. These files contain information related to the administration tools’ settings and history. You must specify a fully-qualified directory name that does not end in a delimiter. See “Administration tools configuration” [SQL Anywhere Server - Programming].

- **New limits for the Timeout, SendBufferSize, and ReceiveBufferSize TCP protocol options** When using TCP/IP, the following protocol options now have upper limits.
  - ReceiveBufferSize (RCVBUFSZ) protocol option The maximum size that you can specify for a buffer used by the TCP/IP protocol stack is 1 MB.
  - SendBufferSize (SNDBUFSZ) protocol option The maximum size that you can specify for a buffer used by the TCP/IP protocol stack is 1 MB.
  - Timeout (TO) protocol option The maximum number of seconds that you can specify to wait for a response when establishing communications (TCP/IP) is 3600 seconds.

If you specify values above these limits, a connection error is returned. See “Connection error: %1” [Error Messages].

- **Database mirroring enhancements** The -xa server option now supports the following values:
  - Specifying DBN=* means that any database can use the server as an arbiter.
  - You can omit the authentication string so that there is no validation of the authentication string provided by a mirror server.
  - Specifying only one authentication string means that all databases must use that authentication string.

See “-xa database server option” [SQL Anywhere Server - Database Administration].

In a high availability, read-only scale-out system, a copy node can act as the arbiter server. You must upgrade all servers in your database mirroring system to use this feature. See “Using a copy node as an arbiter” [SQL Anywhere Server - Database Administration].
● **Statement performance improvements**  The performance of statements that use stored procedures or user-defined functions has been improved.

● **LogFormat (LF) protocol option indicates unknown HTTP method for URL**  If a web request fails due to an unsupported HTTP request method or a URI that is either malformed or missing a required database name, the HTTP method (@M) and HTTP version (@V) return the string `??` and the URI (@U) returns the given request preceded by `>>`. See “LogFormat (LF) protocol option” [SQL Anywhere Server - Database Administration].

● **New properties**  The following connection properties have been added:
  ○ LastCommitRedoPos

  The following database properties have been added:
  ○ LastCommitRedoPos
  ○ LastWrittenRedoPos
  ○ LastSyncedRedoPos

  The following database server properties have been added:
  ○ ProcessID

● **Improved performance of statements that use stored procedures or user-defined functions**  Improvements have been made to the way the database server processes user-defined procedures and functions to improve performance. This performance improvement depends on the number of procedure and user-defined-function calls.

● **ALTER SERVER statement behavior change**  In previous releases, all clauses of the ALTER SERVER statement cause an automatic commit. The CONNECTION CLOSE clause no longer causes an automatic commit. See “ALTER SERVER statement” [SQL Anywhere Server - SQL Reference].

● **OPENSTRING clause enhancement**  The OPENSTRING clause of the FROM clause now supports the `[ NOT | AUTO ] COMPRESSED` option and the `ENCRYPTED` option. The `ENCRYPTED` option can be used to read files that were created by the UNLOAD statement when the ENCRYPTED clause was specified. See “FROM clause” [SQL Anywhere Server - SQL Reference].

● **Enhancements to Certificate Creation utility (createcert)**  A number of options have been added to the Certificate Creation utility to make it easier to create certificates. See “Certificate Creation utility (createcert)” [SQL Anywhere Server - Database Administration].

● **Enhancements to query plan strings**  Plans now include known values for expressions and long plans also include information about the cached plan for a statement.

  When you use application profiling, long plans now include additional information about how a query was optimize and information about the predicates used in a partial index scan.

● **Query optimization**  An optimization has been added to infer sargable IN predicates, which can be used for partial index scans from OR predicates that cannot be transformed into AND predicates. See “Optimizations performed during query processing” [SQL Anywhere Server - SQL Usage].
Support added for ADO.NET Entity Framework 4.2  SQL Anywhere now provides support for the ADO.NET Entity Framework 4.2 Code First feature. See “Entity Framework support” [SQL Anywhere Server - Programming].

New salocation option for SetupVSPackage .NET assemblies installer  The SetupVSPackage application now allows you to specify the location of the SQL Anywhere installation by using the salocation option. See “.NET client deployment” [SQL Anywhere Server - Programming].

Support added to JDBC 4.0 driver for OSGi bundling  The SQL Anywhere JDBC 4.0 driver (sajdbc4.jar) now contains the proper manifest information to allow it to be loaded as an OSGi (Open Services Gateway initiative) bundle. See “JDBC support” [SQL Anywhere Server - Programming].

SQL Remote behavior changes

SQL Remote supports HTTP/HTTPS as a message system for SQL Remote messages for remote databases. Use the SET REMOTE OPTION command to configure the database.

See:

- “The HTTP message system” [SQL Remote]
- “SET REMOTE OPTION statement [SQL Remote]” [SQL Remote]
- “Tutorial: Setting up a replication system using the HTTP message system” [SQL Remote]
- “Tutorial: Setting up a replication system using the HTTP message system with the consolidated database as the message server” [SQL Remote]
- “Tutorial: Setting up a replication system using the HTTP message system and the consolidated database as the message server via Relay Server” [SQL Remote]

UltraLite behavior changes

- ALTER TABLE...ADD CONSTRAINT requires that not nullable columns be specified when adding a unique constraint  Improvements to the ALTER TABLE statement require that all constraint columns must be not nullable when adding a new unique constraint.

See “ALTER TABLE statement [UltraLite]” [UltraLite - Database Management and Reference].

MobiLink file transfers are now supported by Android smartphones  The DatabaseManager.createFileTransfer method can now be used to transfer files between an Android client and a MobiLink server.

See:

- “DatabaseManager.createFileTransfer method [UltraLiteJ]” [UltraLite - Java Programming]
- “FileTransfer interface [UltraLiteJ]” [UltraLite - Java Programming]
- “FileTransferProgressListener interface [UltraLiteJ]” [UltraLite - Java Programming]
- “FileTransferProgressData interface [UltraLiteJ]” [UltraLite - Java Programming]

ZLIB compression is now supported by Android smartphones  The UltraLiteJ API now supports ZLIB data compression for Android smartphones when synchronizing with a MobiLink server through an HTTPS protocol.

The following methods are available for Android smartphones to allow ZLIB compression:
StreamHTTPParms.setZlibCompression

StreamHTTPParms.setZlibDownloadWindowSize

StreamHTTPParms.setZlibUploadWindowSize

StreamHTTPParms.zlibCompressionEnabled

See:

- “MobiLink client network protocol options” [MobiLink - Client Administration]
- “StreamHTTPParms interface [UltraLiteJ]” [UltraLite - Java Programming]

**End-to-end encryption (E2EE) is now supported by Android smartphones**  The UltraLiteJ API now supports end-to-end encryption for Android smartphones when synchronizing with a MobiLink server through an HTTPS protocol.

The following methods are available for Android smartphones to support E2EE:

- StreamHTTPParms.getE2eePublicKey
- StreamHTTPParms.setE2eePublicKey

See:

- “MobiLink client network protocol options” [MobiLink - Client Administration]
- “StreamHTTPParms interface [UltraLiteJ]” [UltraLite - Java Programming]

**Trusted certificates from the default trusted certificate store for synchronization over the HTTPS protocol are now supported by Android smartphones**  Certificates are used according to the following rules of precedence:

1. If the StreamHTTPParms.setTrustedCertificates method is called, then the certificates from the specified file are used.

2. If the StreamHTTPParms.setTrustedCertificates method is not called and certificates were set in the database by the ulinit or ulload utilities, then those certificates are used.

3. If certificates are not specified by either the StreamHTTPParms.setTrustedCertificates method or by the ulinit or ulload utilities, and you are on Android, then certificates are read from the operating system’s trusted certificate store. This certificate store is used by web browsers when they connect to secure web servers via HTTPS.

See:

- “StreamHTTPParms.setTrustedCertificates method [UltraLiteJ]” [UltraLite - Java Programming]

**Cursor movement methods are now supported by Android smartphones**  The UltraLiteJ API now supports all the SQL result set navigational methods that are available in other UltraLite APIs.
The following navigational methods are now available for Android smartphones:

- ResultSet.afterLast
- ResultSet.beforeFirst
- ResultSet.first
- ResultSet.getRowCount
- ResultSet.last
- ResultSet.relative

See:

- “Row data retrieval” [UltraLite - Java Programming]
- “ResultSet interface [UltraLiteJ]” [UltraLite - Java Programming]

- **Some BlackBerry-specific ResultSet methods are now also supported by Android smartphones**

  The get methods requiring a string parameter that are supported on BlackBerry smartphones are now available for Android smartphones.

  The following methods are now supported by Android:

  - ResultSet.getBlobInputStream(String name)
  - ResultSet.getBoolean(String name)
  - ResultSet.getClobReader(String name)
  - ResultSet.getBytes(String name)
  - ResultSet.getDate(String name)
  - ResultSet.getDecimalNumber(String name)
  - ResultSet.getDouble(String name)
  - ResultSet.getFloat(String name)
  - ResultSet.getLong(String name)
  - ResultSet.getSize(String name)
  - ResultSet.getString(String name)
  - ResultSet.isUUIDValue(String name)

  See “ResultSet interface [UltraLiteJ]” [UltraLite - Java Programming].
ResultSetMetaData methods are now supported by Android smartphones

The following methods are now supported on Android smartphones and return values that closely approximate those returned for Blackberry smartphones.

- ResultSetMetaData.getAliasName(int column_no)
- ResultSetMetaData.getDomainName(int column_no)
- ResultSetMetaData.getCorrelationName(int column_no)
- ResultSetMetaData.getQualifiedName(int column_no)
- ResultSetMetaData.getTableColumnName(int column_no)
- ResultSetMetaData.getTableName(int column_no)
- ResultSetMetaData.getWrittenName(int column_no)

See “ResultSetMetadata interface [UltraLiteJ]” [UltraLite - Java Programming].

The DatabaseInfo.getPageSize method is now supported by Android smartphones

The DatabaseInfo.getPageSize method now queries the UltraLite database to retrieve the page size.

See “DatabaseInfo.getPageSize method [UltraLiteJ]” [UltraLite - Java Programming].

The DatabaseInfo.getNumberRowsToUpload method is now supported by Android smartphones

The DatabaseInfo.getNumberRowsToUpload method now returns the number of rows awaiting upload.

See “DatabaseInfo.getNumberRowsToUpload method [UltraLiteJ]” [UltraLite - Java Programming].

DatabaseInfo.getRelease values returned on Android smartphones now include the build number

The DatabaseInfo.getRelease method returns the full software release number.


Extra MobiLink client network protocol option settings are now supported by Android smartphones

The StreamHTTPParms.getExtraParameters and StreamHTTPParms.setExtraParameters methods have been added to specify and retrieve a semicolon-delimited list of MobiLink client network protocol options, and support extra options that were not previously supported by Android smartphones.

See:

- “StreamHTTPParms.getExtraParameters method [Android] [UltraLiteJ]” [UltraLite - Java Programming]
- “StreamHTTPParms.setExtraParameters method [Android] [UltraLiteJ]” [UltraLite - Java Programming]

Error reporting has improved for the DatabaseManager class UltraLiteJ API

A ULjException error is now thrown with the SQLE_NOT_CONNECTED error code if the connect or createDatabase methods are called after calling the release method.
See “DatabaseManager.release method [UltraLiteJ]” [UltraLite - Java Programming].

- **Restartable HTTP is available for the UltraLiteJ API**  
  When restartable HTTP is enabled, UltraLiteJ can tolerate network interruptions so that synchronizations do not fail as often on unreliable networks.

  See:
  - “StreamHTTPParms.setRestartable method [UltraLiteJ]” [UltraLite - Java Programming]
  - “StreamHTTPParms.isRestartable method [UltraLiteJ]” [UltraLite - Java Programming]

- **Null ResultSet handling has improved for the UltraLiteJ API**  
  Null values retrieved by the ResultSet.getString method are now returned as null, rather than an empty string.

  See “ResultSet.getString method [UltraLiteJ]” [UltraLite - Java Programming].

- **Row limiting on a BlackBerry smartphone always enables lazy load indexes**  
  Databases on BlackBerry smartphones that are accessed with row limiting enabled now always lazy load indexes.

  See:
  - “ConfigPersistent.setRowScoreFlushSize method [BlackBerry] [UltraLiteJ]” [UltraLite - Java Programming]
  - “ConfigPersistent.setRowScoreMaximum method [BlackBerry] [UltraLiteJ]” [UltraLite - Java Programming]

- **Secure synchronization of a BlackBerry smartphone through a BlackBerry Enterprise Server has improved**  
  Some BlackBerry smartphones that synchronize over the HTTPS protocol with a BlackBerry Enterprise Server (BES) may require the ":EndToEndRequired" term to be added to the URL suffix to avoid a synchronization failure.

  See “StreamHTTPParms.setURLSuffix method [UltraLiteJ]” [UltraLite - Java Programming].

- **Database page sizes can be adjusted with the UltraLite Java Edition Database Load Utility with the -z option**  
  The new -z option can be used to specify the page size, in bytes, of an UltraLite Java edition database.

  See “ UltraLite Java Edition Database Load utility (uljload)” [UltraLite - Database Management and Reference].

- **Connection.getNewUUID and UUID.toString method support on UltraLite for M-Business Anywhere**  
  The Connection.getNewUUID and UUID.toString methods have been reinstated.

- **The qualified name for a table column can be obtained with the UltraLite C++ API**  
  When used in conjunction with the ULResultSetSchema.GetColumnName method, the ul_name_type_qualified identifier of the ul_column_name_type enumeration can obtain the associated qualified name for a column in the ResultSet object.
See:

- “ul_column_name_type enumeration [UltraLite C++]” [UltraLite - C and C++ Programming]
- “ULResultSetSchema.GetColumnName method [UltraLite C++]” [UltraLite - C and C++ Programming]

- **The Pocket PC 2003 platform is now supported by the UltraLite C++ API** UltraLite is now supported on the Pocket PC 2003 platform. Support for TLS and HTTPS synchronization encryption is not available.

- **Xcode 4.2 support added to CustDB and Names samples** The CustDB and Names samples are now supported by Xcode 3.2 and 4.2.

### MobiLink new features

- **NetworkData class added to MobiLink server APIs** A new NetworkData class has been added to the Java and .NET scripting APIs to give access to HTTP headers and client-side certificates used in a request.

  See:

  - “NetworkData interface [MobiLink server Java]” [MobiLink - Server Administration]
  - “NetworkData interface [MobiLink server .NET]” [MobiLink - Server Administration]

- **collect_network_data protocol has been added to the mlsrv12 -x option** Use the collect_network_data protocol option with the mlsrv12 -x option to enable synchronization scripts to read network information from each synchronization. See “-x mlsrv16 option” [MobiLink - Server Administration].

### MobiLink behavior changes

- **Updates to non-synchronized columns** Updates to non-synchronized columns (except the timestamp column or logical delete column defined in the synchronization model) no longer update the timestamp column when a trigger is used in the synchronization model to maintain the timestamp.

- **Table mapping direction menus** The table mapping direction menus remain enabled after changing the mapping direction to "Not synchronized". See “Table and column mappings” [MobiLink - Getting Started].

- **-qi option for mlsis and dbmlsync** The -qi option is now included in the usage information for mlsis and dbmlsync. See “-qi dblsn option” [MobiLink - Server-Initiated Synchronization] and “-qi dbmlsync option” [MobiLink - Client Administration].

- **New error messages for mlreplay** The following error messages have been added for the MobiLink replay utility:

  - **MLGENREPLAYAPI_FAILED_TO_DETERMINE_CLIENT_TYPE (-5062)** Unable to determine whether or not the recorded protocol came from a dbmlsync client.

  - **MLREPLAY_INCOMPATIBLE_RECORDED_PROTOCOL_VERSION (-5075)** %1 is an incompatible recorded protocol version with this version of mlreplay.
- **MLGENREPLAYAPI_INCOMPATIBLE_RECORDED_PROTOCOL_VERSION (-5076)** is an incompatible recorded protocol version with this version of the Replay API generator.

- **Enhanced dbmlsync logging**  
  Dbmlsync now prints information about the operating system and machine architecture on which it is running to the log. It also prints the platform for which the executable was compiled. See “SQL Anywhere client logs” [MobiLink - Client Administration].

- **Support for SHA2 certificates**  
  MobiLink server and MobiLink clients now support certificates signed using SHA2. See “trusted_certificates” [MobiLink - Client Administration] and “-x mlsrv16 option” [MobiLink - Server Administration].

### Relay server new features

- **Relay Server quick deployment on Windows IIS6, IIS7, and Apache**  
  Relay Server provides a set of utilities that walk you through the deployment process for Windows IIS 6.0, Windows IIS 7.0 or 7.5, and Apache. For more information, see:
  - “Deploying the Relay Server components to Microsoft IIS 6.0 on Windows Server 2003” [Relay Server]
  - “Deploying the Relay Server components to Microsoft IIS 7.0, 7.5, or 8.0” [Relay Server]
  - “Deploying the Relay Server components to Apache on Linux” [Relay Server]

- **Relay Server supports identity forwarding**  
  The SAP Gateway provides several means of authenticating clients, including X.509 certificate forwarding through trusted intermediaries. Relay Server can forward identity information from the remote client and forward it to the SAP NetWeaver Gateway or Web Dispatcher using HTTP headers. See “Backend farm section properties” [Relay Server].

### SQL Anywhere new features

Following is a list of new features in SQL Anywhere version 12.0.1.

For information about changes to the list of supported platforms, see [http://www.sybase.com/detail?id=1061806](http://www.sybase.com/detail?id=1061806).

- **Enhancements to support for spatial data**  
  The following enhancements have been made to the spatial data support in SQL Anywhere:

  **Improved performance**  
  Substantial performance improvements have been made for the following spatial operations:

  - Loading time for shapefiles
  - Loading of polygons and multipolygons and collections containing polygons belonging to round-Earth spatial reference systems
  - Loading of complex polygons and multipolygons (for example, polygons defined by many rings or rings containing many points)
Spatial set operations such as ST_Union and ST_Intersection applied to complex geometries

Spatial predicates such as ST_Contains and ST_Intersects

Spatial predicates where one of the geometries is a point

Spatial predicates applying ST_WithinDistance or ST_Distance to indexed round-Earth geometry columns

**Interactive SQL: Import Wizard now supports shapefiles**  
The Import Wizard now includes an option for importing ESRI shapefiles. See “Importing data with the Import Wizard” [SQL Anywhere Server - SQL Usage].

**Interactive SQL: New FORMAT SHAPEFILE and SRID clauses for the INPUT statement**  
The INPUT statement now supports loading ESRI shapefiles using the FORMAT SHAPEFILE clause. The SRID clause has also been added to the INPUT statement for specifying the SRID to use when loading the shapefile. See “INPUT statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference].

**New st_geometry_load_shapefile system procedure**  
The st_geometry_load_shapefile system procedure allows you to load an ESRI shapefile by providing the name of the file, the SRID to use for loading the data, and the name of the table to create and load the data into. The columns for the table are taken from the column names specified in the shapefile. See “st_geometry_load_shapefile system procedure” [SQL Anywhere Server - SQL Reference].

**Note**  
You must upgrade your database to access this new stored procedure.

**New database option to specify ST_CircularString interpolation tolerance**  
The st_geometry_interpolation option has been added to control the interpolation of ST_CircularString geometries. See “st_geometry_interpolation option” [SQL Anywhere Server - Database Administration].

**Enhancements to ST_WithinDistanceFilter**  
The spatial predicate ST_WithinDistanceFilter is now supported for geometries in round-Earth spatial reference systems. See “ST_WithinDistanceFilter method for type ST_Geometry” [SQL Anywhere Server - Spatial Data Support].

**Enhancement to java_class_path option**  
You can now use the java_class_path database option to specify the classes and JAR files that the system class loader should add to the classpath before launching the Java VM. This option is useful when your application must provide the directories or JAR files. See “java_class_path option” [SQL Anywhere Server - Database Administration].

**New MATVIEW ODBC connection parameter**  
Using the MATVIEW connection parameter, you can specify the string to return for the table type of materialized views when the ODBC function SQLTables runs. By default, the value that is returned by the SQLTables function is VIEW. See “MatView (MATVIEW) connection parameter” [SQL Anywhere Server - Database Administration].

**New -kp database server option**  
You can now use the -kp option to specify the server principal in the standard form of server-name/hostname@REALM. The -kp option enables Kerberos
authenticated connections to the database server. See “-kp database server option” [SQL Anywhere Server - Database Administration].

- **PartnerState property enhancements**  The PartnerState property now returns one of the following values when used with the DB_PROPERTY function:
  - **connected**  There is a connection from the current server to the specified server and a connection from the specified server to the current server.
  - **incoming only**  There is a connection from the specified server to this server.
  - **outgoing only**  There is a connection from this server to the specified server.
  - **disconnected**  There are no connections between this server and the specified server.
  - **NULL**  The database is not mirrored.

See “PartnerState database property” [SQL Anywhere Server - Database Administration].

- **MirrorServerState property enhancements**  The MirrorServerState property now returns one of the following values when used with the DB_EXTENDED_PROPERTY function:
  - **connected**  There is a connection from the current server to a specified server and a connection from the specified server to the current server.
  - **incoming only**  There is a connection from the specified server to this server.
  - **outgoing only**  There is a connection from this server to the specified server.
  - **disconnected**  There are no connections between this server and the specified server.
  - **NULL**  The database is not mirrored.

See “MirrorServerState database property” [SQL Anywhere Server - Database Administration], “DB_PROPERTY function [System]” [SQL Anywhere Server - SQL Reference], and “DB_EXTENDED_PROPERTY function [System]” [SQL Anywhere Server - SQL Reference].

- **New sp_forward_to_remote_server procedure**  You can use the sp_forward_to_remote_server stored procedure to allow an application to execute a SQL statement on a remote server and retrieve result sets generated by the statement. See “sp_forward_to_remote_server system procedure” [SQL Anywhere Server - SQL Reference].

  **Note**
  You must upgrade your database to access this stored procedure.

- **New sa_user_defined_counter_add system procedure**  You can use the sa_user_defined_counter_add system procedure to change the value of a user-defined property. See sa_user_defined_counter_add system procedure. See “sa_user_defined_counter_add system procedure” [SQL Anywhere Server - SQL Reference].
Note
You must upgrade your database to access this stored procedure.

- **New sa_user_defined_counter_set system procedure** You can use the sa_user_defined_counter_set system procedure to set the value of a user-defined property. See “sa_user_defined_counter_set system procedure” [SQL Anywhere Server - SQL Reference].

Note
You must upgrade your database to access this stored procedure.

- **SQLANYSAMP12 environment variable** The Unix and Mac OS X installer now sets the SQLANYSAMP12 environment variable in the sa_config and sample_config scripts. On Unix, the sample_config script may be used to create a per-user copy of the samples. This is useful for a multi-user installation. For single-user installations, the sa_config script sets SQLANYSAMP12 to $SQLANY12/samples. See “SQLANYSAMP16 environment variable” [SQL Anywhere Server - Database Administration] and “Sourcing sa_config.sh and sample_config32.sh/sample_config64.sh [Unix and Mac OS X]” [SQL Anywhere Server - Database Administration].

- **IN parameters of data types LONG VARCHAR, LONG BINARY, and LONG NVARCHAR are now allowed in remote procedure calls** A remote procedure call can now contain IN parameters of data types LONG VARCHAR, LONG NVARCHAR, and LONG BINARY. In addition, parameters of data types VARCHAR, NVARCHAR, and BINARY are no longer restricted to 255 bytes. See “Creating remote procedures (Sybase Central)” [SQL Anywhere Server - SQL Usage].

- **USING clause of the CREATE SERVER statement can now contain variables** The USING clause of a CREATE SERVER statement can now contain variables. This feature allows users to create dynamic remote data access servers. See “CREATE SERVER statement” [SQL Anywhere Server - SQL Reference].

- **Remote data access can load the SQL Anywhere ODBC driver directly** You can define a remote server so that remote data access loads the SQL Anywhere driver directly, bypassing the ODBC driver manager on both Windows and Unix platforms. When defining the remote server, use the syntax below, followed by the remainder of the connection string:

  ```sql
  CREATE SERVER remote-server CLASS 'SAODBC' USING 'DRIVER=SQL Anywhere Native;...';
  ```

  If there are multiple SQL Anywhere remote servers defined without using 'DRIVER=SQL Anywhere Native', then remote data access still uses a driver manager for the other remote servers.

- **AT clause of the CREATE EXISTING TABLE statement and CREATE PROCEDURE statement can now contain variables** The AT clause for the CREATE EXISTING TABLE and CREATE PROCEDURE statements can now contain variables. This feature allows users to map a proxy table or proxy procedure to multiple remote tables or procedures. See “CREATE EXISTING TABLE statement” [SQL Anywhere Server - SQL Reference] and “CREATE PROCEDURE statement” [SQL Anywhere Server - SQL Reference].
• **Enhancement to the RAISERROR clause of the MERGE statement and to the RAISERROR statement**  By using the RAISERROR statement, the SQL Anywhere database server allows an application to raise a customized error. The database server also provides a built-in global variable, SQLCODE, whose value can be examined to determine the specific error raised during the execution of the last statement on the current connection. The database server now reports the user-specified error number for SQLCODE instead of a fixed -631 error message. See “MERGE statement” [SQL Anywhere Server - SQL Reference] and “RAISERROR statement” [SQL Anywhere Server - SQL Reference].

• **VALIDATE TEXT INDEX statement**  You can use the VALIDATE TEXT INDEX statement to verify that the positional information for the terms in the text index is intact. If the positional information is corrupted, an error is generated. See “VALIDATE statement” [SQL Anywhere Server - SQL Reference].

• **Extended syntax for TOP and LIMIT clauses**  The `TOP { ALL | limit } START AT startat` and `LIMIT limit [ OFFSET offset ]` clauses now support simple arithmetic expressions for the `limit`, `offset`, and `startat` arguments. TOP supports the ALL limit, indicating that all rows are to be returned after the specified `startat` value. The maximum value for `(limit + offset)` and `(limit + startat -1)` has been increased to $922372036854775807 = 2^{64}-1$.

  See:
  
  o  “Row limitation clauses in SELECT, UPDATE, and DELETE query blocks” [SQL Anywhere Server - SQL Usage]
  o  “SELECT statement” [SQL Anywhere Server - SQL Reference]
  o  “DELETE statement” [SQL Anywhere Server - SQL Reference]
  o  “UPDATE statement” [SQL Anywhere Server - SQL Reference]

• **SQL Anywhere OLE DB provider now supports the DBTYPE_DBTIMESTAMPOFFSET data type**  The SQL Anywhere OLE DB provider now supports the `DBTYPE_DBTIMESTAMPOFFSET` data type. `DBTYPE_DBTIMESTAMPOFFSET` (146) is an OLE DB type that supports the `TIMESTAMP WITH TIME ZONE` (or `DATETIMEOFFSET`) data type. Support for this data type facilitates transfer of data tables between SQL Anywhere databases and other database management systems (including SQL Anywhere).

• **Web services now supports improved control of HTTP redirect operation**  The new SET REDIR clause of the CREATE PROCEDURE and CREATE FUNCTION statements provides control of the maximum number of re-directions allowed and specifies which HTTP statuses to automatically redirect.

  A web service procedure specifying a POST HTTP method that receives a 303 status issues a redirect request using the GET HTTP method.

  HTTP client procedures now handle relative path re-directions. Previously, re-directions would only succeed if the server had provided an absolute URL.

  A GET method receiving a redirect only provides the query component as specified by the redirect response's location header URL. A POST method receiving a redirect, issues a request URL containing the path and query components as specified by the redirect response location header. Its body contains the query component as generated by the procedure.
Query parameters can now be specified both within the URL clause and (automatically generated) from parameters passed to a procedure. This only applies to procedures specifying a GET HTTP method.

- **Secure web services now supported on Windows Mobile**  
  Web service procedures that use HTTPS and HTTPS_FIPS are now supported on Windows Mobile.

- **SQL Anywhere JDBC driver now supports PreparedStatement.setClob()**  
  The SQL Anywhere JDBC driver now supports PreparedStatement.setClob().

  In previous releases, the SQL Anywhere JDBC driver provided support for PreparedStatement.setBlob, ResultSet.getBlob, and ResultSet.getClob. The SQL Anywhere JDBC driver now supports two of the three PreparedStatement.setClob methods. These are listed below.

  ```java
  PreparedStatement.setClob(int parameterIndex, Clob x)
  PreparedStatement.setClob(int parameterIndex, Reader reader, long length)
  ```

  The following variation is not supported by the SQL Anywhere JDBC driver.

  ```java
  PreparedStatement.setClob(int parameterIndex, Reader reader)
  ```

  When using the `PreparedStatement.setClob(int parameterIndex, Clob x)` overload, the user-supplied Clob implementation only needs to support the Clob.length and Clob.getCharacterStream methods. Also, for large strings and large character streams, the new PreparedStatement.setClob methods within the SQL Anywhere JDBC driver are preferable over the PreparedStatement.setString and PreparedStatement.setCharacterStream methods in terms of both performance and memory usage. See “JDBC support” [SQL Anywhere Server - Programming].

- **SQL Anywhere Java VM ClassLoader now supports shutdown hooks**  
  The SQL Anywhere Java VM ClassLoader that is used in providing Java in the database support allows applications to install shutdown hooks. See “Shutdown hooks in the Java VM” [SQL Anywhere Server - Programming].

- **SQL Anywhere .NET SetupVSPackage installer**  
  The SetupVSPackage application now performs several installer functions such as updating the Global Assembly Cache and the Windows Microsoft.NET `machine.config` file. If SQL Server 2008 or later is installed on the system, SetupVSPackage also installs two mapping files called `MSSqlToSA.xml` and `SAToMSSql10.xml` in the SQL Server `DTS\MappingFiles` folder. See “.NET client deployment” [SQL Anywhere Server - Programming].

### Performance enhancements

Following is a list of performance enhancements introduced in version 12.0.1 for which there are no user-visible changes other than performance improvement.

- **Immediate text indexes**  
  Performance of immediate text index maintenance has improved. The performance improvement you observe will depend on the workload and on the document and text index content. Text indexes indexing long documents benefit the most from this improvement.
• **Physical deletes are deferred to the cleaner** To improve transaction performance, rows are not physically deleted at commit time. Instead, they are left marked as logically deleted with the actual removal being deferred to the cleaner. To prevent excessive database growth, the cleaning schedule has been made more aggressive. See “sa_clean_database system procedure” [SQL Anywhere Server - SQL Reference].

• **Bypassable queries may now have computed columns defined for the table in the FROM clause** One of the conditions for a query to be considered for bypassing the optimizer was to have no computed columns defined for the base table referenced in the FROM clause. This restriction has been removed starting with SA 12.0.1. The common subexpression elimination process using computed columns is done for bypassed queries similar to the queries using the SQL Anywhere optimizer.

• **Bypass queries now possible when base table contains computed columns** Previously, a query could not qualify as a bypass query if the base table referenced in the FROM clause had computed columns. This restriction has been removed. See “Eligibility to skip query processing phases” [SQL Anywhere Server - SQL Usages].

• **Enhancements to the optimizer's use of indexes** Several enhancements have been made to the indexing capabilities of SQL Anywhere, including improvements to optimize queries that use partial index scans and multiple index scans that use the same index.

• **IS NOT NULL search condition** The optimizer can now use an index to evaluate the IS NOT NULL search condition. See “Query predicates” [SQL Anywhere Server - SQL Usage].

---

**Properties and Performance Monitor statistics**

Following is a list of enhancements made to properties and Performance Monitor statistics in SQL Anywhere version 12.0.1.

• **Connection properties** The following connection properties have been added in this release:

  - ConnectedTime
  - UserDefinedCounterRate01
  - UserDefinedCounterRate02
  - UserDefinedCounterRate03
  - UserDefinedCounterRate04
  - UserDefinedCounterRate05
  - UserDefinedCounterRaw01
  - UserDefinedCounterRaw02
  - UserDefinedCounterRaw03
  - UserDefinedCounterRaw04
  - UserDefinedCounterRaw05

  For more information about these properties, see “List of connection properties” [SQL Anywhere Server - Database Administration].
Database properties  The following database properties have been added in this release:

- ApproximateCPUTime
- BytesReceived
- BytesReceivedUncomp
- BytesSent
- BytesSentUncomp
- CarverHeapPages
- ClientStmtCacheHits
- ClientStmtCacheMisses
- ConnectedTime
- HeapsCarver
- HeapsLocked
- HeapsQuery
- HeapsRelocatable
- PacketsReceived
- PacketsReceivedUncomp
- PacketsSent
- PacketsSentUncomp
- PrepStmt
- QueryHeapPages
- QueryMemActiveCurr
- QueryMemGrantFailed
- QueryMemGrantGranted
- QueryMemGrantRequested
- QueryMemGrantWaited
- QueryMemGrantWaiting
- ReqCountActive
- ReqCountBlockContention
- ReqCountBlockIO
- ReqCountBlockLock
- ReqCountUnscheduled
- ReqTimeActive
- ReqTimeBlockContention
- ReqTimeBlockIO
- ReqTimeBlockLock
- ReqTimeUnscheduled
- RequestsReceived
- Rlbk
- UserDefinedCounterRate01
- UserDefinedCounterRate02
- UserDefinedCounterRate03
- UserDefinedCounterRate04
- UserDefinedCounterRate05
- UserDefinedCounterRaw01
- UserDefinedCounterRaw02
- UserDefinedCounterRaw03
- UserDefinedCounterRaw04
● Database server properties

  The following database server properties have been added in this release:

  ○ ApproximateCPUS
  ○ Commit
  ○ ConnectedTime
  ○ Cursor
  ○ CursorOpen
  ○ PrepStmt
  ○ Rlbk
  ○ UserDefinedCounterRate01
  ○ UserDefinedCounterRate02
  ○ UserDefinedCounterRate03
  ○ UserDefinedCounterRate04
  ○ UserDefinedCounterRate05
  ○ UserDefinedCounterRaw01
  ○ UserDefinedCounterRaw02
  ○ UserDefinedCounterRaw03
  ○ UserDefinedCounterRaw04
  ○ UserDefinedCounterRaw05

  For more information about these properties, see “List of database server properties” [SQL Anywhere Server - Database Administration].

● Performance Monitor statistics

  The following Performance Monitor statistics have been added in this release:

  ○ User Defined Rate: Counter1
  ○ User Defined Rate: Counter2
  ○ User Defined Rate: Counter3
  ○ User Defined Rate: Counter4
  ○ User Defined Rate: Counter5
  ○ User Defined Raw: Counter1
  ○ User Defined Raw: Counter2
  ○ User Defined Raw: Counter3
  ○ User Defined Raw: Counter4
  ○ User Defined Raw: Counter5

SQL Anywhere behavior changes

Following is a list of behavior changes to SQL Anywhere introduced in version 12.0.1. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.
Escape characters used by the LOAD and UNLOAD TABLE statements   In previous releases, the escape character for these statements could not be longer than 1 byte. Now, it is recommended that the string you specify for the escape character is no longer than 255 bytes, but it is recommended that the string be no longer than one character.

Minimum and initial cache sizes may increase depending on maximum cache size   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 maximum cache size. Consequently, the minimum and initial cache sizes may often be larger than they were in previous versions.

Changes to the scope of the reserved_keywords option   Previously, you could set the reserved_keywords option for individual users, or specify that the setting had a temporary scope. However, if the temporary or user-level settings differed from the corresponding PUBLIC settings when executing certain DDL statements, you could have encountered problems while recovering or rebuilding the database.

The following behavior change for the reserved_keywords option applies to new version 12.0.1 databases:

○ Temporary and non-PUBLIC settings are not allowed.

The following behavior changes for the reserved_keywords option apply to existing version 12 databases:

○ Existing non-PUBLIC settings are ignored during execution.

○ Removal of existing non-PUBLIC settings is allowed.

○ Dbunload ignores existing non-PUBLIC settings.

Multiprogramming level options   Previously, multiprogramming level database server options that applied strictly to the network database server (dbsrv12) were ignored by the personal database server (dbeng12) without generating an error. Now, the following database server options generate a warning if they are used on the personal database server:

○ -gna dbsrv12 server option
○ -gnh dbsrv12 server option
○ -gnl dbsrv12 server option
○ -gns dbsrv12 server option

Change to the -zt database server option   Setting the -zt database server option no longer makes the ReqCountBlockIO and ReqTimeBlockIO properties available. These two properties are now always available, whether or not the RequestTiming connection property is turned on.

Change to the ODBC connection parameter for the SQLTables function   When you run the ODBC function SQLTables, the TABLE_TYPE column now returns by default the value VIEW for materialized views. Previously, the function would return the value MATERIALIZED VIEW. You can change this default using the MATVIEW connection parameter. See “MatView (MATVIEW) connection parameter” [SQL Anywhere Server - Database Administration].
• **CONVERT function** Parsing of formatted time strings has been enhanced so that the time portion of a string is accepted provided that it matches the format `hh:nn:ss.ssssssAA`. The time string must specify the hour digits, but all other time parts are optional. The AM/PM indicator is always accepted whether or not time parts are omitted. This permits up to six digits to represent microseconds after the seconds.

This change affects the conversion of string to TIME and also to TIMESTAMP. See “CONVERT function [Data type conversion]” [*SQL Anywhere Server - SQL Reference*].

Previously, when converting a string to a time using the CONVERT function with a specified *format-style*, SQL Anywhere 10 and later could have rejected conversions permitted by earlier versions. For example, the following statement is accepted by version 9, but rejected by version 10 and later:

```sql
SELECT CONVERT( TIME, '11:45am', 14 ) tm_conv
```

The behavior of converting strings to TIME changed from version 9 of SQL Anywhere to version 10 and later, with version 10 and later applying the same rules that conversions from string to TIMESTAMP used. For example, the string `11:45am` does not precisely match the format style 14 (`hh:nn:ss.sss`) because it contains an `am` indicator that is not present in the style.

The following statement is accepted by version 9 but rejected by version 10 and later because the string does not match the style format 101 (`mm/dd/yyyy`):

```sql
SELECT CONVERT( TIME, '1991-02-03 11:45', 101 )
```

• **Changing the licensed user and company name with the Server License utility (dblic)** You can now change the licensed user and company without specifying the user count and license type. See “Server Licensing utility (dblic)” [*SQL Anywhere Server - Database Administration*].

---

### SQL Anywhere deprecated and discontinued features

**Note**

As with all forward-looking statements, the lists of deprecated features are not guaranteed to be complete and are subject to change.

The following is a list of deprecated features in version 12.0.1:

• **-kr option** Previously Kerberos server principals needed to be in the form `server-name@default-realm`.

  The Kerberos server principal, including the realm, can be specified with the `-kp` server option. The server principal specified by `-kp` must have been extracted to the Kerberos keytab file on the computer running the database server. The `-kr` option cannot be specified if the `-kp` option is specified. See “-kp database server option” [*SQL Anywhere Server - Database Administration*].

---

Copyright © 2014, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0 93
What's new in version 12.0.1

- **SQL Anywhere JDBC 3.0 driver**  The SQL Anywhere JDBC 3.0 driver is deprecated in this software release. It is recommended that applications using sajdbc.jar switch to sajdbc4.jar. See “JDBC support” [SQL Anywhere Server - Programming].

### MobiLink new features

Following is a list of additions to MobiLink introduced in version 12.0.1. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

### Consolidated databases

Following is a list of enhancements to consolidated database support for MobiLink introduced in SQL Anywhere version 12.0.1.

- **SAP Sybase IQ now supported**  The MobiLink server now supports SAP Sybase IQ as a consolidated database. See “SAP Sybase IQ consolidated database” [MobiLink - Server Administration]. For information about recommended drivers, see http://www.sybase.com/detail?id=1011880.

- **Support for ASE 15.5 added**  The MobiLink server on Windows and Linux now supports consolidated databases running on an ASE 15.5 server. For information about recommended drivers, see http://www.sybase.com/detail?id=1011880.

### MobiLink server

Following is a list of enhancements to the MobiLink server introduced in SQL Anywhere version 12.0.1.

- **Automatic adjustment of database worker threads**  The MobiLink server can now automatically adjust the number of database worker threads to maximize throughput. See “-wm mlsrv16 option” [MobiLink - Server Administration].

- **New named system parameters**  The following new named system parameters have been added to MobiLink so that scripts can identify a new remote ID or user name:
  - **new_remote_id**  This new system parameter indicates a new remote ID and is available in the authenticate_user and begin_synchronization connection scripts, and the begin_synchronization table script.
  - **new_username**  This new system parameter indicates a new user name and is available in the authenticate_user and begin_synchronization connection scripts, and the begin_synchronization table script.
For more information, see:

- “authenticate_user connection event” [MobiLink - Server Administration]
- “begin_synchronization connection event” [MobiLink - Server Administration]
- “begin_synchronization table event” [MobiLink - Server Administration]

- **bigtime and bigdatetime data types added to support ASE 15.5** These data types should be mapped to TIME and TIMESTAMP respectively in SQL Anywhere and UltraLite. See “Adaptive Server Enterprise data mapping” [MobiLink - Server Administration].

- **Maximum length for authentication parameters has been extended** Each MobiLink authentication parameter can be up to 4000 bytes long. The previous limit was 128 bytes. See “Authentication parameters” [MobiLink - Server Administration].

**New mslrv12 features**

- **New -wm option to enable automatic adjustment of database worker threads** The -wm mslrv12 option has been added. Use the -wm option to set the maximum number of database worker threads, allowing the MobiLink server to automatically choose the optimal number of database worker threads. See “-wm mlsrv16 option” [MobiLink - Server Administration].

- **The -cmax, -cmin and -cinit options now support using percentage for the size parameter** The size parameter for the -cmax, -cmin, and -cinit server options can now be specified as a percentage. See “-cmax mlsrv16 option” [MobiLink - Server Administration], “-cmin mlsrv16 option” [MobiLink - Server Administration], and “-cinit mlsrv16 option” [MobiLink - Server Administration].

- **New log_bad_request parameter for -x mlsrv12 option** The -x mlsrv12 option now supports the log_bad_request option for HTTP, HTTPS, and OE. When set to yes, the MobiLink server prints an error if it receives an incomplete or unexpected HTTP request. See “-x mlsrv16 option” [MobiLink - Server Administration].

- **DER encoded keys are now supported for end-to-end encryption** The e2ee_private_key option for the -x mlsrv12 server option now supports both PEM and DER encoded keys.

**New MobiLink central administration of remote databases features**

Following is a list of enhancements to the MobiLink central administration of remote databases feature introduced in SQL Anywhere version 12.0.1.

- **New variables for task conditions** The MobiLink Agent for central administration of remote databases now supports two new variables for task conditions. They are: {is_online} and {network_conn_name}. The variable {is_online} can be used to check whether the client device is connected to a network on which the MobiLink server of the agent can be reached. The variable {network_conn_name} evaluates to the name of the network connection that is used by the Agent for communication with its MobiLink server. See “Variables in parameters” [MobiLink - Server Administration].

- **New support for random delay of task execution** A remote task can now have a random delay interval, which is an interval set in seconds. Each Agent uses the interval to generate a random
number of seconds to delay each task execution. See “Remote tasks” [MobiLink - Server Administration].

- The MobiLink Agent for central administration of remote databases can now be configured interactively

The MobiLink Agent can now be configured interactively on a remote device through a series of configuration windows.

Server utilities

Following is a list of enhancements to the MobiLink server utilities introduced in SQL Anywhere version 12.0.1.

MobiLink replay utility (mlreplay)

- The mlreplay utility can run multiple simulated clients to replay a recorded protocol using only the command line

You control the number of simulated clients using the -n mlreplay option. See “MobiLink Replay utility (mlreplay)” [MobiLink - Server Administration].

- New -rep mlreplay option added

The -rep mlreplay option specifies the number of times simulated clients should replay the recorded protocol. See “MobiLink Replay utility (mlreplay)” [MobiLink - Server Administration].

- New -rnt mlreplay option added

The -rnt mlreplay option instructs simulated clients to start new repetitions of protocol replays until the given runtime is reached. See “MobiLink Replay utility (mlreplay)” [MobiLink - Server Administration].

- New @data mlreplay and mlgenreplayapi option added to support configuration files

The @data mlreplay option allows you to run mlreplay and mlgenreplayapi with the command line specifications stored in a configuration file. See “MobiLink Replay utility (mlreplay)” [MobiLink - Server Administration], “MobiLink Generated Replay API utility (mlgenreplayapi)” [MobiLink - Server Administration], and “Configuration files” [SQL Anywhere Server - Database Administration].

- New -ls mlreplay option added

The -ls mlreplay option logs statistics for each simulated client. See “MobiLink Replay utility (mlreplay)” [MobiLink - Server Administration].

- New -os mlreplay and mlgenreplayapi option added

The -os option limits the maximum size of a log file. See “MobiLink Replay utility (mlreplay)” [MobiLink - Server Administration] and “MobiLink Generated Replay API utility (mlgenreplayapi)” [MobiLink - Server Administration].

- Multiple simulated clients can now be run at the command line

You can now use the asterisk character in your -u, -p, and -r parameters to specify multiple usernames, passwords, and remote IDs, respectively, at the command line. The asterisk character is substituted with the simulated client number for each simulated client call.

The new -rp mlreplay option allows you to change the asterisk wildcard symbol to a different symbol. You can then use the new symbol to specify multiple usernames, passwords, and remote IDs. See “MobiLink Replay utility (mlreplay)” [MobiLink - Server Administration].
MobiLink plug-in for Sybase Central

Following is a list of enhancements to the MobiLink plug-in for Sybase Central introduced in SQL Anywhere version 12.0.1.

- **Synchronization model deployment now creates synchronization profiles**  Now when deploying a synchronization model to a remote database, a synchronization profile is created automatically.

- **Synchronization model support for SAP Sybase IQ**  Synchronization models can now be created with SAP Sybase IQ consolidated databases. See “SAP Sybase IQ consolidated database” [MobiLink - Server Administration].

- **Timestamp downloads using column defaults instead of triggers**  Synchronization models using SQL Anywhere, SAP Sybase IQ, IBM DB2 LUW, and MySQL consolidated databases can now use timestamp-based downloads without having to use triggers to maintain TIMESTAMP columns. Instead of triggers, column default values can be used to automatically update the TIMESTAMP column in a row when the row is inserted or updated. To facilitate this, a new **Use Column Default Instead Of A Trigger** option has been added to the **Timestamp Download Options** page of the **Create Synchronization Model Wizard** and the **Download Type** tab of the table mapping editor.

- **Remote schema names can be imported from a consolidated database**  When you add a consolidated database to a project, either with the **Create Project Wizard** or the **Add Consolidated Database Wizard**, the wizard automatically checks if there are any remote schema names defined in the consolidated database that are not already in the project. If there are, you are asked if you want to import them.

- **Groups can be created from a file**  The **Create Group Wizard** allows you to select the Agents for a group by pointing to a text file that contains the names of the Agents. Members can also be adding by browsing the Agents defined in a consolidated database.

- **New help button for remote task commands**  There is now a question mark icon to the right of the **Command Type** dropdown list that you can use to display help for the commands used to create remote tasks.

- **New Description field for Agents**  The **Agent Properties** window and **Create MobiLink Agent Wizard** now have a **Description** field where you can enter a description of the Agent.

- **New CustDB sample project file added**  A CustDB sample project file, `%SQLANY12%MobiLink\CustDB\project.mlp`, is now available so that you can easily work with CustDB projects and view database scripts. For more information, see “CustDB sample for MobiLink” [MobiLink - Getting Started] and “Tutorial: Building the UltraLite CustDB sample application” [UltraLite - Database Management and Reference].

MobiLink clients

Following is a list of enhancements to MobiLink clients introduced in SQL Anywhere version 12.0.1.
- **Automatic resumption of HTTP sessions**  When an HTTP connection is lost, MobiLink clients automatically attempt to reconnect to the MobiLink server and continue the synchronization. The synchronization only fails if the client cannot reconnect to the MobiLink server after a few attempts.

- **DER encoded keys are now supported for end-to-end encryption**  The e2ee_public_key client protocol option now supports both PEM and DER encoded keys.

- **ECC curve support has been extended to support 15 curves**  The supported ECC curves are: sect163k1, sect163r2, sect233k1, sect233r1, sect283k1, sect283r1, sect409k1, sect409r1, sect571k1, sect571r1, secp192r1, secp224r1, secp256r1, secp384r1, and secp521r1.

- **Support for public keys from x.509 certificates**  End-to-end encryption now supports public keys from x.509 certificates. See “e2ee_public_key” [MobiLink - Client Administration].

### Relay Server

The following Relay Server features have been added in this release:

- **The Relay Server Outbound Enabler (RSEO) now supports HTTP authentication (Basic and Digest) against web servers and HTTP proxies**  The following new (optional) network connection options have been added to the Outbound Enabler:
  - **http_userid**  Userid for authentication.
  - **http_password**  Password for authentication.
  - **http_proxy_userid**  Userid for proxy authentication.
  - **http_proxy_password**  Password for proxy authentication.
  - **proxy_host**  Specifies the host name or IP address of the proxy server.
  - **proxy_port**  Specifies the port number of the proxy server.

For more information, see:

- “Outbound Enabler” [Relay Server]
- “MobiLink client network protocol options” [MobiLink - Client Administration]

### MobiLink behavior changes

Following is a list of behavior changes to MobiLink introduced in version 12.0.1. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

### MobiLink server changes

- **32-bit MobiLink server is no longer supported on 64-bit operating systems**  On 64-bit operating systems, the MobiLink server is only installed as a 64-bit application and must be run as a
64-bit application. Any 32-bit versions of the MobiLink server that might remain on your system from previous version 12 installations cannot be run on a 64-bit system.

**Note**

For MobiLink server versions prior to 12.0.1, an error is not issued if the MobiLink server is run as a 32-bit application on a 64-bit operating system.

As of version 12.0.1, error -10381 is returned if the MobiLink server is run as a 32-bit application on a 64-bit operating system. See “The MobiLink Server must be run as a 64-bit application on a 64-bit operating system” [Error Messages].

- **New behavior for -w mlsrv12 option** If you use the -wm option to automatically adjust the number of database worker threads, the -w option for mlsrv12 now sets the initial number of database worker threads. Previously, this option set the number of database worker threads. See “-w mlsrv16 option” [MobiLink - Server Administration] and “-wm mlsrv16 option” [MobiLink - Server Administration].

- **The -cm option is now an alias for the -cmax option** The -cm option is an alias for the -cmax server option. See “-cmax mlsrv16 option” [MobiLink - Server Administration].

**MobiLink server utilities changes**

Following is a list of behavior changes to MobiLink server utilities introduced in version 12.0.1.

**MobiLink replay utility (mlreplay)**

- **New behavior for -o and -ot mlreplay and mlgenreplayapi options** The -o and -ot options for mlreplay and mlgenreplayapi now log command line options. See “MobiLink Replay utility (mlreplay)” [MobiLink - Server Administration] and “MobiLink Generated Replay API utility (mlgenreplayapi)” [MobiLink - Server Administration].

- **The -n and -sci mlreplay options can now be used together under certain conditions** The -n and -sci mlreplay options can be used together when the number of simulated clients specified by the -n option is less than or equal to the number of simulated clients in the simulated client information file. When used together, -n specifies the number of simulated clients run.

These options allow one simulated client information file, specifying x number of simulated clients, to replay a protocol with 1 to x simulated clients. The old behavior only allowed you to replay recorded protocol with exactly x number of simulated clients. See “MobiLink Replay utility (mlreplay)” [MobiLink - Server Administration].

- **Code template now available for IdentifySimulatedClient and FiniIdentifySimulatedClient callbacks** When generated, these callbacks contain commented out code that you can implement when using a generic username, password, and remote ID for a replay session. See “IdentifySimulatedClient callback” [MobiLink - Server Administration] and “FiniIdentifySimulatedClient callback” [MobiLink - Server Administration].
MobiLink client changes

Following is a list of behavior changes to MobiLink clients introduced in version 12.0.1.

- The following changes have been made to the DBMLSync Setup For SQL Anywhere window
  - The Retry on remote progress, Remote is behind, Remote is ahead, Drop conflicting connections, Site Script, and Cmdline help options have been removed.
  - The Publication option has been replaced with a Subscription option.
  - If you attempt to change your MobiLink password, the software verifies that you have entered a non-empty password and that the new password and the verify password match before allowing you to close the window.
  - When the window is opened after a synchronization attempt because an incorrect MobiLink password was entered, a message is displayed indicating what the problem was. Previously, the error message was displayed in the dbmlsync window.

- New default for persistent HTTP and HTTPS synchronizations
  By default MobiLink clients (except UltraLiteJ) synchronizing via HTTP or HTTPS now use persistent connections. To return to the previous behavior, clients should specify persistent=no in the client network protocol options.

MobiLink plug-in for Sybase Central changes

Following is a list of behavior changes for the MobiLink plug-in for Sybase Central introduced in version 12.0.1.

- The MobiLink User and Subscription page of the Deploy Synchronization Model Wizard has changed
  The MobiLink User and Subscription page of the Deploy Synchronization Model Wizard has been renamed to the MobiLink User and Synchronization Profile page. The page now has an option to specify the synchronization profile name. The synchronization profile is now created when a synchronization model is deployed.

- Synchronization model deployment option to choose defaults appropriate for remote tasks
  The first page of the Deploy Synchronization Model Wizard now has an option when deploying to a remote database to initialize the wizard with settings appropriate for creating remote tasks for central administration of remote databases. This option is selected by default.

MobiLink deprecated and discontinued features

Note
As with all forward-looking statements, the lists of deprecated features are not guaranteed to be complete and are subject to change.
IBM DB2 mainframe no longer supported  IBM DB2 mainframe is no longer supported as a consolidated database. However, MobiLink still supports DB2 LUW (Linux, Unix, and Windows) as a consolidated database.

SQL Remote new features

Following is a list of additions to SQL Remote introduced in version 12.0.1. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

New control parameters for SET REMOTE OPTION statement  When SQL Remote is running in continuous mode and an error occurs when accessing the message system, you can now control:

- the number of times you want SQL Remote to retry the send and/or receive phases before shutting down. See the max_retries common option in “SET REMOTE OPTION statement [SQL Remote]” [SQL Anywhere Server - SQL Reference].
- the amount of time to wait between retries. See the pause_after_failure common option in “SET REMOTE OPTION statement [SQL Remote]” [SQL Anywhere Server - SQL Reference].

SQL Remote behavior changes

Following is a list of changes to SQL Remote introduced in version 12.0.1.

Improved performance dealing with missing messages  Prior to SQL Remote version 16.0, if the consolidated database terminated before all the messages in a multi-part message were received, the remote database would delete all the received messages and send a resend request to the consolidated database. The consolidated database would then scan the applicable transaction log to satisfy the resend request. With SQL Remote version 16.0, on startup, SQL Remote will generate a warning message about missing messages, delete the existing messages, and ask for a resend only if it has seen a message that contains a commit and that is the last part of the multi-part message, or that is the next message after the previous multi-part message.

UltraLite new features

Following is a list of additions to UltraLite introduced in version 12.0.1. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

General features

UltraLite is now supported on the Android smartphone  UltraLite now supports Android smartphones. You can use the Android implementation of the UltraLiteJ API to create applications that use UltraLite databases. The implementation usage differs from the BlackBerry implementation, which only supports UltraLite databases that are specific to BlackBerry smartphones. For more information, see “Benefits of UltraLite APIs for Windows Mobile” [UltraLite - Database].
A new tutorial demonstrates how to create an UltraLite application on an Android smartphone or simulator in the Eclipse environment. The tutorial is based on a new code sample located in the `%SQLANY12%UltraLite\Android\CustDB` directory. See “Tutorial: Building an Android application” [UltraLite - Java Programming].

**Tutorial for building a BlackBerry application now supports the Eclipse environment**  The BlackBerry tutorial has been updated to use the Eclipse environment. See “Tutorial: Building a BlackBerry application” [UltraLite - Java Programming].

**Dynamic cache sizing**  UltraLite now grows its database file cache when warranted in response to database operations and as available memory allows. Applications can also resize the cache explicitly (typically in cases where the application is requested to reduce its memory usage). See:

- “UltraLite CACHE_SIZE connection parameter” [UltraLite - Database Management and Reference]
- “UltraLite CACHE_MIN_SIZE connection parameter” [UltraLite - Database Management and Reference]
- “UltraLite CACHE_MAX_SIZE connection parameter” [UltraLite - Database Management and Reference]
- “UltraLite cache_allocation option” [UltraLite - Database Management and Reference]

**Support for download-only tables in UltraLite**

*_download_only suffix_  For UltraLite databases (not UltraLite Java edition databases), changes to the table on the consolidated database are downloaded during synchronization, but local changes are not sent up to MobiLink. Any table with the suffix _download_only is marked as download-only. Tables can also be set to download-only by specifying SYNCHRONIZE DOWNLOAD for the synchronization constraint clause of the CREATE TABLE and ALTER TABLE SQL statements. See “UltraLite download-only tables” [UltraLite - Database Management and Reference], “CREATE TABLE statement [UltraLite]” [UltraLite - Database Management and Reference], and “ALTER TABLE statement [UltraLite]” [UltraLite - Database Management and Reference].

**New TABLE_IS_DOWNLOAD_ONLY flag**  Uncommitted client-side changes may result in download conflicts when tables are synchronized. For UltraLite and UltraLite Java edition, the table_flag column in the systable system table may now include the TABLE_IS_DOWNLOAD_ONLY flag.

**sync=download attribute for tables**  The XML format for an unloaded UltraLite database now also includes the sync="download" attribute on tables. See “systable system table” [UltraLite - Database Management and Reference].

**TLS identities can be stored in the database**  For UltraLite databases (not UltraLite Java edition databases), TLS identities, consisting of an X.509 certificate, a private key, and, optionally, a chain of certificates of the certificate authorities that signed the client's certificate, can now be stored in the UltraLite database at creation time. New options have been added to the ulinit and ulload utilities. See “UltraLite Initialize Database utility (ulinit)” [UltraLite - Database Management and Reference].
● **UltraLiteJ code sample locations are now organized by device and platform support**  
  Code samples are now located in the following directories:

  ○ For BlackBerry samples: `%SQLANY12%\UltraLiteJ\BlackBerry`
  ○ For Java J2ME samples: `%SQLANY12%\UltraLiteJ\J2ME`
  ○ For Java J2SE samples: `%SQLANY12%\UltraLiteJ\J2SE`
  ○ For Android samples: `%SQLANY12%\UltraLiteJ\Android`

  See “Code examples” [UltraLite - Java Programming].

---

### Platforms and devices

#### New devices

Android smartphones are now supported by the UltraLiteJ programming interface. The UltraLiteJ programming interface is common to the BlackBerry and Android platforms, and applications must be written in Java for each platform. On BlackBerry smartphones, the underlying DBMS is a Java implementation of UltraLite, while on Android, the DBMS is the C++ version that is also provided on iPhone and iPad, Windows Mobile, and Windows.

See:

  ○ “UltraLite overview” [UltraLite - Database Management and Reference]
  ○ “UltraLiteJ application development” [UltraLite - Java Programming]
  ○ “Tutorial: Building an Android application” [UltraLite - Java Programming]
  ○ “UltraLiteJ API reference” [UltraLite - Java Programming]

---

### Programming interfaces

#### UltraLite.NET

#### Asynchronous synchronization

The following members have been added to support asynchronous synchronization:

  ○ `public delegate void ULSyncProgressedDlg( IAsyncResult result, ULSyncProgressData data )`  
  This method is invoked during synchronization to provide progress information. See “ULSyncProgressedDlg delegate [UltraLite.NET]” [UltraLite - .NET Programming].

  ○ `public IAsyncResult ULConnection.BeginSynchronize()`  
  This method creates a new thread to perform a synchronization and then returns immediately. See “ULConnection.BeginSynchronize method [UltraLite.NET]” [UltraLite - .NET Programming].

  ○ `public IAsyncResult ULConnection.BeginSynchronize( Control control, ULSyncProgressedDlg dlg, object state )`  
  This method creates a new thread to perform a
synchronization and then returns immediately. See “ULConnection.BeginSynchronize method [UltraLite.NET]” [UltraLite - .NET Programming].

- **public void ULConnection.CancelSynchronize( IAsyncResult asyncResult )** This method tells the synchronization thread to terminate and then returns immediately. See “ULConnection.CancelSynchronize method [UltraLite.NET]” [UltraLite - .NET Programming].

- **public void ULConnection.EndSynchronize( IAsyncResult asyncResult )** This method blocks until an asynchronously launched synchronization terminates. A ULException is thrown if the synchronization failed. See “ULConnection.EndSynchronize method [UltraLite.NET]” [UltraLite - .NET Programming].

- **public bool ULSyncProgressData.IsFinalSyncProgress** This property is true if this is the final synchronization progress message for a synchronization. See “ULSyncProgressData.IsFinalSyncProgress property [UltraLite.NET]” [UltraLite - .NET Programming].

### UltraLite for M-Business Anywhere

This API is deprecated for UltraLite versions 12.0.0 and 12.0.1.

### UltraLiteJ

- **Database configuration support for Android devices** The new `DatabaseManager.CreateConfigurationFileAndroid` method establishes the `ConfigFileAndroid` object for a persistent database saved in a file on an Android device. See “ConfigFileAndroid interface [Android] [UltraLiteJ]” [UltraLite - Java Programming].

- **Android-related updates to the ConfigPersistent interface** The following members are affected by the UltraLiteJ support for Android:

<table>
<thead>
<tr>
<th>Member name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>enableAesDBEncryption method</td>
<td>New for Android only. Enables AES encryption of the database. See “ConfigPersistent.enableAesDBEncryption method [UltraLiteJ]” [UltraLite - Java Programming].</td>
</tr>
<tr>
<td>getAutoCheckpoint method</td>
<td>Deprecated for BlackBerry; active for Android.</td>
</tr>
<tr>
<td>getConnectionString method</td>
<td>New for Android only. Gets a connection string registered with SetConnectionString. See “ConfigPersistent.getConnectionString method [Android] [UltraLiteJ]” [UltraLite - Java Programming].</td>
</tr>
<tr>
<td>getCreationString method</td>
<td>New for Android only. Gets a creation string registered with SetCreationString. See “ConfigPersistent.getCreationString method [Android] [UltraLiteJ]” [UltraLite - Java Programming].</td>
</tr>
</tbody>
</table>
### Member name | Status
--- | ---
getDatabaseKey method | New for Android only. Gets a database encryption key registered with setEncryptionKey. See “ConfigPersistent.getEncryptionKey method [UltraLiteJ]” [UltraLite - Java Programming].

getUserName method | New for Android only. Gets the name of a user set by setUserName. See “ConfigPersistent.getUserName method [Android] [UltraLiteJ]” [UltraLite - Java Programming].

setAutocheckpoint method | Deprecated for BlackBerry; active for Android.

setConnectionString method | New for Android only. Sets the connection string to be used to create or connect to a database. See “ConfigPersistent.setConnectionString method [Android] [UltraLiteJ]” [UltraLite - Java Programming].

setCreationString method | New for Android only. Sets the creation string to be used to create a database. See “ConfigPersistent.setCreationString method [Android] [UltraLiteJ]” [UltraLite - Java Programming].

setDatabaseKey method | New for Android only. Sets the key for encryption. See “ConfigPersistent.setEncryptionKey method [UltraLiteJ]” [UltraLite - Java Programming].

setEncryption method | BlackBerry only.

setUserName method | New for Android only. Sets the name of the user. See “ConfigPersistent.setUserName method [Android] [UltraLiteJ]” [UltraLite - Java Programming].

For more information, see “ConfigPersistent interface [UltraLiteJ]” [UltraLite - Java Programming].

### Android-related updates to the Connection interface
The following members are affected by the UltraLiteJ support for Android:

<table>
<thead>
<tr>
<th>Member name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkpoint method</td>
<td>Deprecated for BlackBerry; active for Android.</td>
</tr>
<tr>
<td>emergencyShutdown method</td>
<td>Not available for Android.</td>
</tr>
<tr>
<td>getDatabaseId method</td>
<td>Not available for Android.</td>
</tr>
<tr>
<td>getDatabaseProperty method</td>
<td>Not available for Android.</td>
</tr>
<tr>
<td>getState method</td>
<td>Not available for Android.</td>
</tr>
<tr>
<td>Member name</td>
<td>Status</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------------------------</td>
</tr>
<tr>
<td>isSynchronizationDeleteDisabled method</td>
<td>Not available for Android.</td>
</tr>
<tr>
<td>OPTION_BLOB_FILE_BASE_DIR variable</td>
<td>BlackBerry and Java SE only.</td>
</tr>
</tbody>
</table>

For more information, see “Connection interface [UltraLiteJ]” [UltraLite - Java Programming].

- **Android-related updates to the DatabaseManager class**  
  The following members are not available for Android:
  - createConfigurationNonPersistent method
  - createFileTransfer method
  For more information, see “DatabaseManager class [UltraLiteJ]” [UltraLite - Java Programming].

- **Android-related updates to the ResultSet interface**  
  The following members are not available for Android:
  - getBoolean method
  - getBytes method
  - getClobReader method
  - getDate method
  - getDecimalNumber method
  - getDouble method
  - getFloat method
  - getLong method
  - getSize method
  - getString method
  - getUUIDValue method
  - isNull method
  For more information, see “ResultSet interface [UltraLiteJ]” [UltraLite - Java Programming].

---

**UltraLite behavior changes and deprecated features**

Following is a list of deprecated features and behavior changes to UltraLite introduced in version 12.0.1. For information about supported platforms and versions, see [http://www.sybase.com/detail?id=1061806](http://www.sybase.com/detail?id=1061806).
Behavior changes

- **Change to SET OPTION behavior** UltraLite now performs a commit when persistent options are set. See “SET OPTION statement [UltraLite]” [UltraLite - Database Management and Reference].

- **Default setting for the persistent network protocol option is now true** The default value for the persistent network protocol option for MobiLink clients (except UltraLiteJ) is now "true". Synchronizations using persistent connections are faster than non-persistent connections, particularly for HTTPS. However, if an intermediary server requests non-persistent connections, or the client detects that an intermediary does not support persistent connections, it will automatically downgrade to non-persistent for the rest of the sync session. See “persistent” [MobiLink - Client Administration].

- **SQLE_COMMUNICATIONS_ERROR (-85) error message changed to SQLE_MOBILINK_COMMUNICATIONS_ERROR (-1305)** All information about the communication problem is now contained in the error. You no longer need access these values from the separate stream-error object; however, existing code that does reference them will continue to work. The parameters for this error are the stream error code, parameters, and system code. See “Communication error” [Error Messages].

- **SQLE_UNKNOWN_PROPERTY error message** SQLE_UNKNOWN_PROPERTY is now signaled for unknown property names. See “‘%1’ is an unknown property” [Error Messages].

- **SQLE_MOBILINK_AUTHENTICATION_FAILED error message** SQLE_MOBILINK_AUTHENTICATION_FAILED is signaled rather than SQLE_INVALID_LOGON during synchronization, and includes the authentication status and values. See “The synchronization failed because MobiLink returned authentication status '%1' with value '%2'” [Error Messages].

**Administration tools new features**

Following is a list of additions to administration tools introduced in version 12.0.1. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

- **OEMs can specify the preference directory used by DBISQL, Sybase Central, DBCConsole, and ML Monitor** When deploying an OEM.ini file along with the administration tools, ensure that the following lines are included in the file to specify which directory is used to store administration tools preferences:

```
[prefeferences]
directory=preferences_file_directory
```

See “Administration tools configuration” [SQL Anywhere Server - Programming].
Sybase Central plug-in new features

Following is a list of additions to Sybase Central plug-ins introduced in version 12.0.1.

SQL Anywhere plug-in new features

- **New application profiling recommendation**  When you use the Application Profiling Wizard to check the overall performance of your database based on its schema, the wizard checks the order of columns in tables. The wizard makes recommendations when the majority of wide columns precede the narrow columns. Ordering columns so that wider columns precede narrow columns can negatively impact response times. Narrow columns should be defined in the table declaration before wider columns, unless they are accessed infrequently.

  Wide columns are columns greater than 15 bytes in size, or LONG data types (for example, LONG VARCHAR), or columns defined as XML. See “Tip: Review the order of columns in tables” [SQL Anywhere Server - SQL Usage].

- **Enhancements to how user permissions are managed and displayed**  Numerous enhancements have been made to the way user permissions are displayed in the plug-in. It is easier to view and change the permissions that have been explicitly set for a database object. When you select a database object, the permission information now appears in a tab in the right pane. Column permissions are displayed with their corresponding table permissions. See “Granting an object-level privilege (Sybase Central)” [SQL Anywhere Server - Database Administration].

Sybase Central plug-in behavior changes

Following is a list of behavior changes to Sybase Central plug-ins introduced in version 12.0.1.

SQL Anywhere plug-in new features

- **Sybase Central only supports SQL Anywhere versions 10x and later databases**  Support for version 9 database servers and databases created with version 9 software has been removed from the SQL Anywhere plug-in. When unloading and reloading the database into a reload file, or into a new or existing database, you can still connect to a database created with version 5, 6, 7, 8 or 9 software running on a version 9 or later database server. See “SQL Anywhere Server upgrades” on page 272.

Interactive SQL new features

Following is a list of additions to Interactive SQL introduced in version 12.0.1.

- **Connect window supports SAP Sybase IQ**  You can use Interactive SQL to connect to an SAP Sybase IQ database. In the Connect window, click Change Database Type, and then click SAP Sybase IQ.

- **Text Completion**  The following list describes changes related to the text completion feature in Interactive SQL and Sybase Central:
○ By default, the text completion window now opens automatically when you type in the SQL Statements pane. You can also open the text completion window by clicking Edit » Open Text Completer or pressing Ctrl+Space.

○ By default, SQL keywords and completed database object names are enclosed in double quotes.

○ SQL statements and keywords are now included in the list of suggestions.

○ The text completion window now handles databases where strings are treated as case-sensitive, but identifiers are not.

○ The following keyboard shortcuts have changed:

<table>
<thead>
<tr>
<th>Old shortcut</th>
<th>New shortcut</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tab</td>
<td>Ctrl+A</td>
<td>Shows a context-free list of matches.</td>
</tr>
<tr>
<td>+</td>
<td>Ctrl+Plus Sign (+)</td>
<td>Adds the item with its parameter list to the SQL Statements pane.</td>
</tr>
<tr>
<td>*</td>
<td>Ctrl+Asterisk</td>
<td>Adds the item with its parameter and type list to the SQL Statements pane.</td>
</tr>
<tr>
<td>'</td>
<td>Ctrl+Double quote (&quot;)</td>
<td>Adds the item enclosed in quotation marks to the SQL Statements pane.</td>
</tr>
<tr>
<td>N/A</td>
<td>Tab</td>
<td>Accepts the selection and closes the text completion window.</td>
</tr>
</tbody>
</table>

For a complete list of shortcuts, see “Text completion keyboard shortcuts” [SQL Anywhere Server - Database Administration].

For more information, see “Text completion” [SQL Anywhere Server - Database Administration].

- **New Import/Export option** When you click Options » Import/Export, you can specify how nulls should be exported in the Export NULL values as field. See “output_nulls option [Interactive SQL]” [SQL Anywhere Server - Database Administration].

## Interactive SQL behavior changes

Following is a list of changes to Interactive SQL introduced in version 12.0.1.

- **READ statement encoding algorithm changed** When running the READ statement in Interactive SQL, the encoding that is used to read the data is now determined in the following order:

  1. The encoding specified by the ENCODING clause (if this clause is specified).

  2. The encoding specified by the byte order mark in the file (if byte order mark is specified).
3. The default TEXT format encoding (if the format encoding is specified).

4. The default encoding for the platform you are running on. On English Windows computers, the default encoding is 1252.

See “READ statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference].

SQL Anywhere Monitor new features

Following is a list of additions to SQL Anywhere Monitor introduced in version 12.0.1.

- **The Monitor can now send email alerts through SMTP servers**  The SQL Anywhere Monitor now supports sending email alerts through SMTP servers that require TLS connections (for example, GMail). See “Enabling the Monitor to send alert emails” [SQL Anywhere Server - Database Administration].

- **Monitor databases in read-only scale-out and in database mirroring systems**  The Monitor supports monitoring the database server of the root node in a read-only scale-out system and the primary database server in a mirroring system. In a mirroring system, the Monitor can be configured to always monitor the primary database server, even when a fail-over occurs. A new widget, the SQL Anywhere Scale-Out Topology widget, displays topology information for the SQL Anywhere mirroring and scale-out systems.

To create a resource for a root node in a read-only scale-out system or the primary database in a mirroring system, use the Add Resources window and choose to create a SQL Anywhere Server resource. See “Adding a database resource” [SQL Anywhere Server - Database Administration].

There is no difference in the resource configuration for a root node in a read-only scale-out system and other database servers. When the resource is configured or migrated, the Monitor detects that the resource is the root node in a read-only scale-out system.

A resource for a primary database in a mirroring system requires special configuration. To ensure that the Monitor always monitors the primary database server even when a fail-over occurs, you must configure the resource using the following options:

- In the Host field, specify the host names and port numbers of the computers running the primary and mirror servers in a comma-separated list. For example: my-primary-server:2638,my-mirror-server:49152.

- Ensure that the Port field is empty.

- In the Server field, type the alternate server name for the primary server. That is, specify the name that clients use to connect to the database server that is acting as the primary server in the database mirroring system.

You must add the SQL Anywhere Scale-Out Topology to your dashboard as it is not one of the default widgets that appears when a new dashboards is created. See “Widgets” [SQL Anywhere Server - Database Administration].
See “Lesson 9: (Optional) Monitoring your read-only scale-out system from the SQL Anywhere Monitor” [SQL Anywhere Server - Database Administration] and “Lesson 8: (Optional) Monitoring a database mirroring system” [SQL Anywhere Server - Database Administration].

Also, a new alert, Alert when the number of disconnected scale-out nodes exceeds the given threshold:, has been added. See “Specifying alert thresholds” [SQL Anywhere Server - Database Administration].

- **New Monitor metrics and alert for MobiLink servers** The Monitor metrics for MobiLink server resources now includes a metric named Is Consolidated Available. This metric shows True if the MobiLink server can connect to the consolidated database. If the consolidated database becomes unavailable, then a Consolidated Database Unavailable alert is issued.

- **New names for MobiLink resource metrics** The following SQL Anywhere Monitor metrics for MobiLink server resources have changed:

<table>
<thead>
<tr>
<th>Old name</th>
<th>New name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bytes Read Rate</td>
<td>TCP Bytes Read Rate</td>
</tr>
<tr>
<td>Bytes Written Rate</td>
<td>TCP Bytes Written Rate</td>
</tr>
</tbody>
</table>

- **Alert Notification window** The Alert Notification window now includes two options that restrict the number of emails sent per day and the type of email sent.

  - **Only Send Email Notifications For High-Priority Alerts** Set this option to receive email notifications only when high-priority alerts are raised.

  - **Do Not Send More Than The Given Number Of Email Notifications Per Day** Specify this option to limit the amount of notification emails sent to each user per day.

  See “Enabling the Monitor to send alert emails” [SQL Anywhere Server - Database Administration].

- **Delete Alerts window** When the user selects an alert and clicks the Delete button, the Delete Alerts window appears, prompting the user to select one of the following options:

  - **Delete Selected Alerts** Choose this option to only delete alerts selected in the Alert List widget.

  - **Delete Alerts Received Before** Select this option to delete all alerts received before a specified time.

  - **Delete Alerts For The Following Resource** Choose this option to delete all alerts for the selected resource.

  See “Deleting alerts” [SQL Anywhere Server - Database Administration].

**SQL Anywhere Monitor behavior changes**

Following is a list of changes to SQL Anywhere Monitor introduced in version 12.0.1.
• **Maximum amount of data that can be exported**  The maximum amount of data that you can export to a file is 25 metrics or 1 million points. See “Exporting metrics” [SQL Anywhere Server - Database Administration].

## Documentation enhancements

Following is a list of changes made to the SQL Anywhere documentation in version 12.0.1.

• **List of SQL Anywhere tutorials**  A complete list of SQL Anywhere tutorials is now included in the documentation. See “List of SQL Anywhere tutorials” [SQL Anywhere 16 - Introduction].
What's new in version 12.0.0

For information about new features and behavior changes in versions of SQL Anywhere before version 10, see http://dcx.sybase.com/html/dbwnen10/dbwnen10.html.

Product-wide new features

Following is a list of product-wide additions introduced in version 12.0.0. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

- **Escape characters supported for configuration files** The parsing for configuration files has been enhanced to support \\ as an escape sequence for a \, and " as an escape sequence for a " See “Configuration file escape characters” [SQL Anywhere Server - Database Administration].

- **Deployment Wizard now supports 64-bit deployments** Previous versions of the Deployment Wizard supported 32-bit deployments. Now 64-bit deployments are supported as well. See “The Deployment Wizard” [SQL Anywhere Server - Programming].

Product-wide behavior changes

Following is a list of product-wide behavior changes introduced in version 12.0.0. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

- **SQL Anywhere Replication Agent for Sybase Replication Server unsupported** The SQL Anywhere Replication Agent for Sybase Replication Server is not supported in version 12. You must use an alternative replication or synchronization technology such as MobiLink or SQL Remote. See “MobiLink technology” [MobiLink - Getting Started] and “SQL Remote systems” [SQL Remote].

  The following changes have been made to the software as a result of this change:

  - **a_change_log DBTools structure** The ignore_ltm_trunc member is no longer supported.

  - **LTMGeneration database property** This property is reserved for system use.

  - **LTMTTrunc database property** This property is reserved for system use.

  - **Log Transfer Manager utility (dbltm)** This utility has been removed.

  - **Log Translation utility (dbtran)** The -is option no longer supports the value RepServer.

    The -rsu option has been removed.

  - **Service utility (dbsvc)** You can no longer create services for Replication Agent. The SQLANYLTM service group is no longer supported.

    The -w and -t options no longer support the value dbltm.
○ Support utility (dbsupport)  This utility no longer returns information for the SQL Anywhere Replication Agent (dbltm).

○ Transaction Log utility (dblog)  The -g and -il options are no longer supported. See “Transaction Log utility (dblog)” [SQL Anywhere Server - Database Administration].

○ replicate_all database option  This option has been removed.

○ delete_old_logs database option  This option is not supported for use with Replication Agent.

○ ALTER PROCEDURE statement  The following syntax is no longer supported:

\[
\text{ALTER PROCEDURE [ owner.]}\text{procedure-name}
\text{REPLICATE \{ ON | OFF \}}
\]

See “ALTER PROCEDURE statement” [SQL Anywhere Server - SQL Reference].

○ ALTER TABLE statement  The REPLICATE \{ ON | OFF \} clause is no longer supported. See “ALTER TABLE statement” [SQL Anywhere Server - SQL Reference].

**Executables now respect the user's umask settings when running as a daemon**  In previous releases when an executable ran as a daemon (it was started with the -ud option) on Unix, the executable ignored the user's umask setting and called umask(0), which created new files with group +other read/write permissions. When you start a SQL Anywhere 12 executable as a daemon, the executable does not call umask(0) and respects the user's umask setting. Because the current user's umask setting controls the permissions for executables, you should ensure that the user's umask value is set to the desired level before starting the executable.

This behavior change applies to the following executables:

○ dbeng12
○ dbsrv12
○ dbitm
○ dbmlsync
○ dbns12
○ dbremote
○ mlsrv12
○ uleng12

---

**SQL Anywhere new features**

Following is a list of new features in SQL Anywhere version 12.0.0.

For information about changes to the list of supported platforms, see [http://www.sybase.com/detail?id=1061806](http://www.sybase.com/detail?id=1061806).
Main features

Following is a list of the main features introduced in SQL Anywhere version 12.0.0.

- **New spatial data support**  The following features have been added in support of the new spatial data capabilities in SQL Anywhere 12.0.0. You must upgrade your database to use this feature.

  **Note**
  Spatial data support for 32-bit Windows and 32-bit Linux requires a CPU that supports SSE2 instructions. This support is available with Intel Pentium 4 or later (released in 2001) and AMD Opteron or later (released in 2003).

**SQL statements**  The following SQL statement enhancements have been made in support of the spatial feature:

- **SHAPEFILE clause**  A new SHAPEFILE format option is available for the OPENSTRING subclause of the FROM clause. See “FROM clause” [SQL Anywhere Server - SQL Reference].

  As well, a new SHAPEFILE format option is available for the FORMAT clause of the LOAD TABLE statement. See “LOAD TABLE statement” [SQL Anywhere Server - SQL Reference].

- **CREATE SPATIAL REFERENCE SYSTEM statement**  Creates or replaces a spatial reference system. See “CREATE SPATIAL REFERENCE SYSTEM statement” [SQL Anywhere Server - SQL Reference].

- **ALTER SPATIAL REFERENCE SYSTEM statement**  Changes the settings of an existing spatial reference system. See the Remarks section for considerations before altering a spatial reference system. See “ALTER SPATIAL REFERENCE SYSTEM statement” [SQL Anywhere Server - SQL Reference].

- **DROP SPATIAL REFERENCE SYSTEM statement**  Drops a spatial reference system. See “DROP SPATIAL REFERENCE SYSTEM statement” [SQL Anywhere Server - SQL Reference].

- **CREATE SPATIAL UNIT OF MEASURE statement**  Creates or replaces a spatial unit of measurement. See “CREATE SPATIAL UNIT OF MEASURE statement” [SQL Anywhere Server - SQL Reference].

- **DROP SPATIAL UNIT OF MEASURE statement**  Drops a spatial unit of measurement. See “DROP SPATIAL UNIT OF MEASURE statement” [SQL Anywhere Server - SQL Reference].

**Interactive SQL changes**  A new viewer tool, the **Spatial Viewer**, has been added to Interactive SQL to allow you to view spatial geometries. You can query spatial data in the top portion of the viewer, and then see your results represented as an image in the lower portion of the viewer. See “Viewing spatial data as images (Interactive SQL)” [SQL Anywhere Server - Spatial Data Support].

Also, when viewing a result row in Interactive SQL, you can now preview a geometry as a Scalable Vector Graphic (SVG) using the new **Spatial Preview** tab. See “Viewing spatial data as images (Interactive SQL)” [SQL Anywhere Server - Spatial Data Support].
New data types, methods, and constructors  New types, methods, and constructors have been added to allow you access, model, and analyze spatial data. See “Accessing and manipulating spatial data” [SQL Anywhere Server - Spatial Data Support].

As well, many spatial compatibility functions have been provided to mimic regular SQL functions when accessing and manipulating spatial data. These functions have been provided for compatibility with other products, and make use of the spatial methods and constructors provided in SQL Anywhere. See “Spatial compatibility functions” [SQL Anywhere Server - Spatial Data Support].

New functions and system procedures  The following functions and system procedures have been added in support of spatial data in the database:

- **TREAT function**  Allows you to change the declared type of a geometry expression to a subtype. See “TREAT function [Data type conversion]” [SQL Anywhere Server - SQL Reference].

- **sa_describe_shapefile system procedure**  Describes the names and types of columns contained in an ESRI shapefile. This system feature is for use with the spatial feature. See “sa_describe_shapefile system procedure” [SQL Anywhere Server - SQL Reference].

- **sa_install_feature system procedure**  Installs additional features that were not present in the database when SQL Anywhere was installed. See “sa_install_feature system procedure” [SQL Anywhere Server - SQL Reference].

- **st_geometry_dump system procedure**  Expands a geometry object into a result set with each row representing one of the geometry objects contained in the input. See “st_geometry_dump system procedure” [SQL Anywhere Server - SQL Reference].

Wizards  In Sybase Central, the following wizards have been added in support of the spatial data feature:

- **Create Spatial Reference System Wizard**  The Create Spatial Reference System Wizard allows you to create new spatial reference systems. See “Creating a spatial reference system” [SQL Anywhere Server - Spatial Data Support].

- **Create Unit Of Measure Wizard**  The Create Unit Of Measure Wizard allows you to create new units of measure for use with spatial data. See “Creating a unit of measure” [SQL Anywhere Server - Spatial Data Support].

Catalog changes  The following changes have been made to the catalog as part of the new spatial data support:

- **SYSSPATIALREFERENCESYSTEM system view**  Each row of the SYSSPATIALREFERENCESYSTEM system view describes a spatial reference system defined in the database. See “SYSSPATIALREFERENCESYSTEM system view” [SQL Anywhere Server - SQL Reference].

- **SYSUNITOFMEASURE system view**  Each row of the SYSUNITOFMEASURE system view describes a unit of measure defined in the database. See “SYSUNITOFMEASURE system view” [SQL Anywhere Server - SQL Reference].
- **ST_GEOMETRY_COLUMNS consolidated view**  Each row of the ST_GEOMETRY_COLUMNS system view describes a spatial column defined in the database. See “ST_GEOMETRY_COLUMNS consolidated view” [SQL Anywhere Server - SQL Reference].

- **ST_SPATIAL_REFERENCE_SYSTEMS consolidated view**  Each row of the ST_SPATIAL_REFERENCE_SYSTEMS system view describes a spatial reference system defined in the database. See “ST_SPATIAL_REFERENCE_SYSTEMS consolidated view” [SQL Anywhere Server - SQL Reference].

- **ST_UNITS_OF_MEASURE consolidated view**  Each row of the ST_UNITS_OF_MEASURE system view describes a unit of measure defined in the database. See “ST_UNITS_OF_MEASURE consolidated view” [SQL Anywhere Server - SQL Reference].

**Database options and properties**  The following database options and properties have been added in support of the spatial data features.

- **st_geometry_asbinary_format option**  Controls how spatial values are converted from a geometry to binary. See “st_geometry_asbinary_format option” [SQL Anywhere Server - Database Administration].

- **st_geometry_astext_format option**  Controls how spatial values are converted from a geometry to text. See “st_geometry_astext_format option” [SQL Anywhere Server - Database Administration].

- **st_geometry_asxml_format option**  Controls how spatial values are converted from a geometry to XML. See “st_geometry_asxml_format option” [SQL Anywhere Server - Database Administration].

- **st_geometry_describe_type option**  Controls how spatial values are described. See “st_geometry_describe_type option” [SQL Anywhere Server - Database Administration].

- **st_geometry_on_invalid option**  Controls the behavior when a geometry fails basic validation. See “st_geometry_on_invalid option” [SQL Anywhere Server - Database Administration].

- **st_geometry_asbinary_format connection property**  Returns a value that indicates how spatial values are converted from a geometry to binary. See “st_geometry_asbinary_format connection property” [SQL Anywhere Server - Database Administration].

- **st_geometry_astext_format connection property**  Returns a value that indicates how spatial values are converted from a geometry to text. See “st_geometry_astext_format connection property” [SQL Anywhere Server - Database Administration].

- **st_geometry_asxml_format connection property**  Returns a value that indicates how spatial values are converted from a geometry to xml. See “st_geometry_asxml_format connection property” [SQL Anywhere Server - Database Administration].

- **st_geometry_describe_type connection property**  Returns a value that indicates how spatial values are described to the client. See “st_geometry_describe_type connection property” [SQL Anywhere Server - Database Administration].
● **st_geometry_on_invalid connection property**  Returns a value that indicates the behavior when a geometry fails basic validation. See “st_geometry_on_invalid connection property” [SQL Anywhere Server - Database Administration].

**SYS_SPATIAL_ADMIN_ROLE group**  Membership in this group allows users to create, alter, or drop spatial reference systems and units of measure.

For more information about SQL Anywhere spatial support, see “Spatial data” [SQL Anywhere Server - Spatial Data Support].

● **Read-only scale-out**  You can now use SQL Anywhere in a read-only scale-out system. In this configuration, one database server (the root node) runs a read-write copy of the database, while other database servers run read-only copies of the database (copy nodes) that can be used to offload reporting and other operations that require read access to the database. Read-only scale-out can be used on its own, or with database mirroring. You must upgrade or rebuild existing databases to use this feature.

A sample has been added in `%SQLANYSAMP12\%SQLAnywhere\DBMirror` that uses a database mirroring system in conjunction with a scale-out system.

See:

○ “SQL Anywhere read-only scale-out” [SQL Anywhere Server - Database Administration]
○ “NodeType (NODE) connection parameter” [SQL Anywhere Server - Database Administration]
○ “CREATE MIRROR SERVER statement” [SQL Anywhere Server - SQL Reference]
○ “ALTER MIRROR SERVER statement” [SQL Anywhere Server - SQL Reference]
○ “SET MIRROR OPTION statement” [SQL Anywhere Server - SQL Reference]
○ “DROP MIRROR SERVER statement” [SQL Anywhere Server - SQL Reference]
○ MIRROR SERVER clause: “COMMENT statement” [SQL Anywhere Server - SQL Reference]
○ “SYSMIRROROPTION system view” [SQL Anywhere Server - SQL Reference]
○ “SYSMIRRORSERVER system view” [SQL Anywhere Server - SQL Reference]
○ “SYSMIRRORSERVEROPTION system view” [SQL Anywhere Server - SQL Reference]
○ “MirrorRole database property” [SQL Anywhere Server - Database Administration]
○ MirrorServerState and MirrorState properties: “sa_mirror_server_status system procedure” [SQL Anywhere Server - SQL Reference]

● **Database mirroring enhancements**  You can now set up a database mirroring system using SQL statements instead of specifying mirroring settings on the database server command line. You must upgrade or rebuild existing databases to use this feature.
See:

- “CREATE MIRROR SERVER statement” [SQL Anywhere Server - SQL Reference]
- “ALTER MIRROR SERVER statement” [SQL Anywhere Server - SQL Reference]
- “SET MIRROR OPTION statement” [SQL Anywhere Server - SQL Reference]
- “DROP MIRROR SERVER statement” [SQL Anywhere Server - SQL Reference]
- MIRROR SERVER clause added to the “COMMENT statement” [SQL Anywhere Server - SQL Reference]
- “SYSMIRROROPTION system view” [SQL Anywhere Server - SQL Reference]
- “SYSMIRRORSERVER system view” [SQL Anywhere Server - SQL Reference]
- “SYSMIRRORSERVEROPTION system view” [SQL Anywhere Server - SQL Reference]
- “MirrorRole database property” [SQL Anywhere Server - Database Administration]
- Extended database properties: MirrorServerState, MirrorState
- “sa_mirror_server_status system procedure” [SQL Anywhere Server - SQL Reference]
- “Database mirroring behavior changes and deprecated features” on page 142

- **Host connection parameter**  The new Host connection parameter takes a host name (or IP address) and optional port number that tells the client where to find the database server. This connection parameter is now the recommended way to connect to database servers running on a different computer than the client. See “Host connection parameter” [SQL Anywhere Server - Database Administration].

- **Enhancements to automatic statistics management**  SQL Anywhere 12 includes a statistics governor that improves the automatic maintenance of statistics on database columns. The health and usefulness of each statistic in the database is automatically evaluated, and required maintenance is performed so that the statistics are self-monitored and self-healing. Statistics maintenance is performed in the background and does not create a significant load on database server performance. See “How the statistics governor maintains statistics” [SQL Anywhere Server - SQL Usage].

The sa_server_option system procedure now supports the following options to help you manage statistics: DropBadStatistics, DropUnusedStatistics, and StatisticsCleaner. See “sa_server_option system procedure” [SQL Anywhere Server - SQL Reference].

- **Sequences**  SQL Anywhere now supports the generation of sequences. Sequences can be used by applications to generate unique key values. Using sequence values can help applications prevent concurrency and performance issues.

You can also create, edit, and manage sequences using the SQL Anywhere plug-in for Sybase Central. For example, use the Create Sequence Generator Wizard to create a new sequence in the database.
See also:

- “Use of a sequence to generate unique values” [SQL Anywhere Server - SQL Usage]
- “CREATE SEQUENCE statement” [SQL Anywhere Server - SQL Reference]
- “ALTER SEQUENCE statement” [SQL Anywhere Server - SQL Reference]
- “DROP SEQUENCE statement” [SQL Anywhere Server - SQL Reference]
- “SYSSEQUENCE system view” [SQL Anywhere Server - SQL Reference]
- “SYSSEQUENCERPERM system view” [SQL Anywhere Server - SQL Reference]
- SEQUENCE clause, “COMMENT statement” [SQL Anywhere Server - SQL Reference]
- GRANT USAGE ON SEQUENCE syntax, “GRANT statement” [SQL Anywhere Server - SQL Reference]
- REVOKE USAGE ON SEQUENCE syntax, “REVOKE statement” [SQL Anywhere Server - SQL Reference]

You must upgrade or rebuild existing databases to use sequences.

- **Multiprogramming level enhancements** The network database server (dbsrv12) now automatically controls its multiprogramming level by default. This behavior allows the database server to improve its throughput and adapt to workload changes without DBA intervention.

When the database server starts, it creates a pool of workers that are used to service requests. The number of workers is the current multiprogramming level of the server. The pool has minimum and maximum limits, and the current multiprogramming level is always within those limits. The DBA can change the minimum and maximum values at start up by using database server options or while the database server is running by using the sa_server_option system procedure.

The following options have been added to allow you to control the database server's multiprogramming level:

<table>
<thead>
<tr>
<th>Database server option</th>
<th>sa_server_option value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>“-gn database server option” [SQL Anywhere Server - Database Administration]</td>
<td>CurrentMultiProgrammingLevel</td>
<td>Sets the initial multiprogramming level of the database server.</td>
</tr>
<tr>
<td>“-gna database server option” [SQL Anywhere Server - Database Administration]</td>
<td>AutoMultiProgrammingLevel</td>
<td>Turns on and off dynamic tuning of the database server’s multiprogramming level.</td>
</tr>
<tr>
<td>“-gnh database server option” [SQL Anywhere Server - Database Administration]</td>
<td>MaxMultiProgrammingLevel</td>
<td>Sets the maximum number of tasks that the database server can execute concurrently.</td>
</tr>
<tr>
<td>“-gnl database server option” [SQL Anywhere Server - Database Administration]</td>
<td>MinMultiProgrammingLevel</td>
<td>Sets the minimum number of tasks that the database server can execute concurrently.</td>
</tr>
</tbody>
</table>
For more information about the multiprogramming level in SQL Anywhere, see “Database server configuration of the multiprogramming level” [SQL Anywhere Server - Database Administration].

- **Immediate materialized views now support outer joins** Materialized views containing OUTER JOINs in their definitions can now be declared immediate. See “Materialized views restrictions” [SQL Anywhere Server - SQL Usage].

- **Selecting from DML statements** You can now specify a DML statement in the FROM clause of the SELECT statement. This feature allows you to write SQL queries over a derived table populated by the rows modified by an UPDATE, INSERT, DELETE, or MERGE statement and return values from these updated rows to the application.

  The most common use of this feature is to verify or validate the values of rows that have been modified by the application. Previously, the only way to accomplish this would be through the use of a trigger and multiple SQL statements. See “FROM clause” [SQL Anywhere Server - SQL Reference] and “SELECT over a DML statement” [SQL Anywhere Server - SQL Usage].

- **Full text search feature now supports external prefilter and term breaker libraries** A new API has been added to allow you to connect to external prefilter and term breaker libraries when creating and updating full text indexes. This means you can take document formats like XML, PDF, and Word and remove unwanted tags and metadata before indexing their content. Sample term breaker libraries can be used to do language- or application-specific term breaking. The sample prefilter and term breaker libraries are included to help you design your own, or you can use third-party libraries. See “Advanced: External term breaker and prefilter libraries” [SQL Anywhere Server - SQL Usage].

  If Microsoft Office is installed on the system running the database server, then IFilters for Office documents such as Word and Excel are available. If the server has Acrobat Reader installed, then a PDF IFilter is likely available.

  The PREFILTER EXTERNAL NAME clause and TERM BREAKER EXTERNAL NAME clause have been added to the ALTER TEXT CONFIGURATION statement to allow you to specify the name and location of your external libraries. See “ALTER TEXT CONFIGURATION statement” [SQL Anywhere Server - SQL Reference].

  The ISYSTEXTCONFIG system table has been modified to hold information about the entry points and the external libraries used for tokenizing and/or prefiltering. Specifically, the existing prefilter column data type has changed to be a LONG VARCHAR to hold the entry points and library name for an external prefilter library. A new LONG VARCHAR column, external_term_breaker, has been added to hold the entry points and library name for an external term breaker library. See “SYSTEXTCONFIG system view” [SQL Anywhere Server - SQL Reference].
You must upgrade your database to use external prefilter and term breaker libraries.

- **Checksum enhancements**  The database server now determines whether to create write checksums for databases pages (checksums that are created only when the pages are written to disk) based on the database version. By default, version 10 and 11 databases do not have global checksums enabled, and version 12 databases have global checksums enabled. When you start an older database on a version 12 database server, the default behavior of the database server is to enable write checksums. For version 12 databases, the database server's default behavior is to not enable write checksums because by default version 12 databases have global checksums enabled. See “Checksums enabled by default for new databases” on page 140.

You can use the CHECKSUM clause of the START DATABASE statement or the -wc option when starting a database or database server to change the database server behavior for write checksums. See “-wc database option” [SQL Anywhere Server - Database Administration], “-wc database server option” [SQL Anywhere Server - Database Administration], and “START DATABASE statement” [SQL Anywhere Server - SQL Reference].

You can disable checksums for a database using the CHECKSUM clause of the ALTER DATABASE statement. See “ALTER DATABASE statement” [SQL Anywhere Server - SQL Reference].

**Database connections**

Following is a list of enhancements made to database connections in SQL Anywhere version 12.0.0.

- **Temporary connections are named**  Temporary connections are used to perform operations such as running backups or initializing databases. You can get information about temporary connections by using the sa_conn_info system procedure, the sa_conn_list system procedure and the Name and ParentConnection connection properties. See:
  
  - “Temporary connections” [SQL Anywhere Server - Database Administration]
  - “Name connection property” [SQL Anywhere Server - Database Administration]
  - “ParentConnection connection property” [SQL Anywhere Server - Database Administration]
  - “sa_conn_info system procedure” [SQL Anywhere Server - SQL Reference]
  - “sa_conn_list system procedure” [SQL Anywhere Server - SQL Reference]

- **Connection pooling**  The ConnectionPool connection parameter controls the behavior of client connection pooling. See “ConnectionPool (CPOOL) connection parameter” [SQL Anywhere Server - Database Administration] and “Connection pooling” [SQL Anywhere Server - Database Administration].

- **Escape connection parameter supported for ODBC data sources**  By default the ODBC driver uses the tilde (~) as an escape character, but some applications assume that the escape character is the backslash (\). You can use the Escape connection parameter to specify the escape character for your application. See “Escape connection parameter (ODBC)” [SQL Anywhere Server - Database Administration].

  In addition, you can specify the Escape connection parameter in the Connect window.
Backup and recovery

Following is a list of backup and recovery enhancements introduced in SQL Anywhere version 12.0.0.

- **Restoring archive backups created with version 12**  
  As of version 12, you cannot restore archive backups created with version 11 or earlier database servers.

- **Free pages are eliminated in archive backups**  
  By default, archive backups skip some free pages, which can result in smaller and potentially faster backups. Free page elimination has no affect on the back up of transaction log files because transaction log files do not contain free pages. So, databases with large transaction log files may not benefit as much from free page elimination as databases with small transaction log files. You can control this behavior using the BACKUP statement's FREE PAGE ELIMINATION clause or using the Backup Database Wizard. See “BACKUP statement” [SQL Anywhere Server - SQL Reference].

  When you are restoring a strongly-encrypted database that was backed up with free page elimination on, the encryption key for the database must be specified. You can specify the encryption key using the RESTORE DATABASE statement's KEY clause or using the Restore Database Wizard.

Security

Following is a list of security enhancements introduced in SQL Anywhere version 12.0.0.

- **FIPS-certified algorithms now available on 64-bit Windows and 32- and 64-bit Linux operating systems**  
  FIPS-certified algorithms can now be used on 64-bit Windows and 32 and 64-bit Linux operating systems.

  For a list of platforms on which the FIPS-certified algorithms are supported, see “Supported platforms” [SQL Anywhere 16 - Introduction].

  For information about using FIPS-certified algorithms, see “FIPS-certified encryption technology” [SQL Anywhere Server - Database Administration].

Database security

You must upgrade or rebuild your database to make use of these changes. See “SQL Anywhere Server upgrades” on page 272.

- **SYS_SPATIAL_ADMIN_ROLE group**  
  Membership in this group allows users to create, alter, or drop spatial reference systems and units of measure.

Database utilities

Following is a list of enhancements made to database utilities in SQL Anywhere version 12.0.0.


- **Database page sizes can now be specified in kilobytes**  
The Extraction utility (dbxtract), Initialization utility (dbinit), and Unload utility (dbunload) now allow you to specify the database page size in units of bytes or kilobytes. Previously, you could only specify the page size in bytes. See:

  - “Extraction utility (dbxtract)” [SQL Remote]
  - “Initialization utility (dbinit)” [SQL Anywhere Server - Database Administration]
  - “Unload utility (dbunload)” [SQL Anywhere Server - Database Administration]

- **File Hiding utility (dbfhide) enhancements**  
On Windows, the File Hiding utility (dbfhide) now supports using a Windows encryption API to obfuscate configuration files. The utility also supports the -q option so it can be run quietly. See “File Hiding utility (dbfhide)” [SQL Anywhere Server - Database Administration].

- **Server Licensing utility (dblic) enhancements**  
In previous releases if you wanted to change your edition of SQL Anywhere, you had to uninstall the software and then reinstall with the new license key. You can now use the -k option for the Server Licensing utility (dblic) to change your edition of SQL Anywhere when you receive a new license key from Sybase iAnywhere. See “Server Licensing utility (dblic)” [SQL Anywhere Server - Database Administration].

- **Service utility (dbsvc) enhancements**  
When specifying dependencies for services with the -rs option of the dbsvc utility on Windows, you can now specify the display name or the actual service name. See “Service utility (dbsvc) for Windows” [SQL Anywhere Server - Database Administration].

- **Support utility (dbsupport) enhancements**  
The -ac option allows you to add a comment that is included in the error report. The -af option allows you to include a file with the error report submission. See “Support utility (dbsupport)” [SQL Anywhere Server - Database Administration].

- **Unload utility (dbunload) enhancements**  
The dbunload utility now supports the following options:

  - The -kd option unloads a database into a single dbspace. See “Unload utility (dbunload)” [SQL Anywhere Server - Database Administration].
  - The -qr option prevents progress messages from being created and displayed when loading tables and creating indexes. See “Unload utility (dbunload)” [SQL Anywhere Server - Database Administration].

---

**Database options**

Following is a list of enhancements made to database options in SQL Anywhere version 12.0.0.

- **uuid_has_hyphens option**  
The uuid_has_hyphens controls the formatting of unique identifier values when they are converted to strings. This feature was removed in version 11, but is now reinstated. See “uuid_has_hyphens option” [SQL Anywhere Server - Database Administration].

- **blocking_others_timeout option**  
This option specifies the amount of time that another connection can block on the current connection's row and table locks before the current connection is rolled back. This option can be used to prevent a low priority task from blocking other connections for longer than the specified time. See “blocking_others_timeout option” [SQL Anywhere Server - Database Administration].
**http_connection_pool_basesize option**  This option specifies a baseline size for connection pools used by HTTP. See “http_connection_pool_basesize option” [SQL Anywhere Server - Database Administration].

**http_connection_pool_timeout option**  This option specifies the maximum duration that an unused connection may be retained within an HTTP connection pool. See “http_connection_pool_timeout option” [SQL Anywhere Server - Database Administration].

**progress_messages option**  This option controls whether progress messages are sent from the database server to the client. You can set this option as a temporary option for the utility database. See “progress_messages option” [SQL Anywhere Server - Database Administration], and “Allowed statements for the utility database” [SQL Anywhere Server - Database Administration].

By default Interactive SQL shows the progress messages in the Messages pane. You can also set the SQL Anywhere database option progress_messages by clicking Tools » Options » SQL Anywhere » Commands and clicking Show Progress Messages. When selected, this option sets the database progress_messages option to Formatted. When cleared the database progress_messages option is set to Off. By default, the Show Progress Messages option is selected.

**timestamp_with_time_zone_format option**  This option controls how TIMESTAMP WITH TIME ZONE values are converted to strings. See “timestamp_with_time_zone_format option” [SQL Anywhere Server - Database Administration].

**reserved_keywords option**  This option turns on individual keywords. See “reserved_keywords option” [SQL Anywhere Server - Database Administration].

**st_geometry_asbinary_format option**  Controls how spatial values are converted from a geometry to binary. See “st_geometry_asbinary_format option” [SQL Anywhere Server - Database Administration].

**st_geometry_astext_format option**  Controls how spatial values are converted from a geometry to text. See “st_geometry_astext_format option” [SQL Anywhere Server - Database Administration].

**st_geometry_asxml_format option**  Controls how spatial values are converted from a geometry to XML. See “st_geometry_asxml_format option” [SQL Anywhere Server - Database Administration].

**st_geometry_describe_type option**  Controls how spatial values are described. See “st_geometry_describe_type option” [SQL Anywhere Server - Database Administration].

**st_geometry_on_invalid option**  Controls the behavior when a geometry fails basic validation. See “st_geometry_on_invalid option” [SQL Anywhere Server - Database Administration].

### Database server options

Following is a list of enhancements made to database server options in SQL Anywhere version 12.0.0.

**-xm option**  The -xm database server option controls how often the database server checks for new IP addresses. If a computer is connected to a new network or it is disconnected from an existing
network and the -xm option is specified, the database server starts listening on the new network when
the change is detected. See “-xm database server option” [SQL Anywhere Server - Database
Administration].

Properties and Performance Monitor statistics

Following is a list of enhancements made to properties and Performance Monitor statistics in SQL
Anywhere version 12.0.0.

● **Connection properties** The following connection properties have been added in this release:
  ○ blocking_others_timeout
  ○ http_connection_pool_basesize
  ○ http_connection_pool_timeout
  ○ ParentConnection
  ○ Progress
  ○ progress_messages
  ○ QueryRowsFetched
  ○ reserved_keywords
  ○ st_geometry_asbinary_format
  ○ st_geometry_astext_format
  ○ st_geometry_asxml_format
  ○ st_geometry_describe_type
  ○ st_geometry_on_invalid
  ○ timestamp_with_time_zone_format

  For more information about these properties, see “List of connection properties” [SQL Anywhere
Server - Database Administration].

● **Database properties** The following database properties have been added in this release:
  ○ ConnPoolCachedCount
  ○ ConnPoolHits
  ○ ConnPoolMisses
  ○ DriveBus
  ○ DriveModel
  ○ HasTornWriteFix
  ○ HttpConnPoolCachedCount
  ○ HttpConnPoolHits
  ○ HttpConnPoolMisses
  ○ HttpConnPoolSteals
  ○ LastCheckpointTime
  ○ MirrorRole
  ○ QueryRowsFetched
  ○ SynchronizationSchemaChangeActive
  ○ WriteChecksums

  For more information about these properties, see “List of database properties” [SQL Anywhere Server
- Database Administration].
The following database server properties have been added in this release:

- AutoMultiProgrammingLevel
- AutoMultiProgrammingLevelStatistics
- CurrentMultiProgrammingLevel
- IPAddressMonitorPeriod
- IsPortableDevice
- MaxMultiProgrammingLevel
- MinMultiProgrammingLevel
- ObjectType
- ThreadDeadlocksAvoided
- ThreadDeadlocksReported

For more information about these properties, see “List of database server properties” [SQL Anywhere Server - Database Administration].

### System procedures and functions

Following is a list of system procedure and function enhancements added in SQL Anywhere version 12.0.0.

- **New sa_text_index_vocab_nchar system procedure**  
  This new system procedure is for use with NCHAR text indexes and is equivalent to sa_text_index_vocab. See “sa_text_index_vocab_nchar system procedure” [SQL Anywhere Server - SQL Reference].

- **New sa_copy_cursor_to_temp_table system procedure**  
  Copies the contents of a cursor to a temporary table. See “sa_copy_cursor_to_temp_table system procedure” [SQL Anywhere Server - SQL Reference].

- **New sa_describe_cursor system procedure**  
  Returns the name and data types of columns in a cursor. While this information can be retrieved from various client programming interfaces, it was not previously accessible within stored procedures. See “sa_describe_cursor system procedure” [SQL Anywhere Server - SQL Reference].

- **New sa_install_feature system procedure**  
  Installs additional features that were not present in the database when SQL Anywhere was installed. See “sa_install_feature system procedure” [SQL Anywhere Server - SQL Reference].

- **New sa_list_cursors system procedure**  
  Returns a result set with one row for each of the cursors maintained by the database server for this connection. The result set gives the cursor name, a value indicating whether the cursor is currently open, and other meta information. See “sa_list_cursors system procedure” [SQL Anywhere Server - SQL Reference].

- **New sa_mirror_server_status system procedure**  
  This procedure reports the status of copy nodes in the tree below the database server on which the procedure is run. See “sa_mirror_server_status system procedure” [SQL Anywhere Server - SQL Reference].
New sa_reserved_words system procedure  This procedure returns a list of SQL Anywhere reserved words. See “sa_reserved_words system procedure” [SQL Anywhere Server - SQL Reference].

Enhancements to the sa_server_option system procedure  The following options have been added to the sa_server_option system procedure:

○ AutoMultiProgrammingLevel
○ AutoMultiProgrammingLevelStatistics
○ CurrentMultiProgrammingLevel
○ DropBadStatistics
○ DropUnusedStatistics
○ IPAddressMonitorPeriod
○ MaxMultiProgrammingLevel
○ MinMultiProgrammingLevel
○ StatisticsCleaner

See “sa_server_option system procedure” [SQL Anywhere Server - SQL Reference].

Enhancements to the sa_table_page_usage system procedure  When the progress_messages database option is set to Raw or Formatted, progress messages are sent from the database server to the client while this system procedure is running. See “sa_table_page_usage system procedure” [SQL Anywhere Server - SQL Reference].

Enhancements to the xp_read_file system procedure  The xp_read_file system procedure now includes an optional parameter that allows you to specify lazy reads. When you specify this optional parameter and its value is not zero, the file is read and then immediately unlocked. See “xp_read_file system procedure” [SQL Anywhere Server - SQL Reference].

Enhancements to the xp_startsmtp system procedure  The xp_startsmtp system procedure supports four new parameters: trusted_certificates, certificate_company, certificate_unit, and certificate_name. These parameters allow you to send mail using secure SMTP. You must upgrade your database to use this feature. See “xp_startsmtp system procedure” [SQL Anywhere Server - SQL Reference].

Enhancements to the OPENXML operator  The OPENXML string operator now supports USING FILE and USING VALUE clauses that allow you to load data from a file or expression of CHAR, NCHAR, BINARY, or LONG BINARY type, or BLOB string, respectively. See “OPENXML operator” [SQL Anywhere Server - SQL Reference].

New MEDIAN function  Computes the median of a numeric expression for a set of rows. See “MEDIAN function [Aggregate]” [SQL Anywhere Server - SQL Reference].

Enhancement to the HASH function  The HASH function now accepts the CRC32 algorithm type. See “HASH function [String]” [SQL Anywhere Server - SQL Reference].

Enhancements to the BIT_OR, BIT_AND, and BIT_XOR functions  The BIT_OR, BIT_AND, and BIT_XOR functions now support integer values and binary values. Also, BIT_OR, BIT_AND, and BIT_XOR functions can now be used in parallel execution plans.
See:

- “BIT_OR function [Aggregate]” [SQL Anywhere Server - SQL Reference]
- “BIT_AND function [Aggregate]” [SQL Anywhere Server - SQL Reference]
- “BIT_XOR function [Aggregate]” [SQL Anywhere Server - SQL Reference]

**Enhancements to the DATEADD, DATEDIFF, DATEPART, and DATENAME functions** The microsecond date part and the TIMESTAMP WITH TIME ZONE data type are now supported by the DATEADD, DATEDIFF, DATEPART, and DATENAME functions. See:

- “DATEADD function [Date and time]” [SQL Anywhere Server - SQL Reference]
- “DATEDIFF function [Date and time]” [SQL Anywhere Server - SQL Reference]
- “DATEPART function [Date and time]” [SQL Anywhere Server - SQL Reference]
- “DATENAME function [Date and time]” [SQL Anywhere Server - SQL Reference]

**Enhancements to the DB_EXTENDED_PROPERTY function** You can now use the DB_EXTENDED_PROPERTY function with the MirrorServerState and MirrorState properties to determine the synchronization and connection status of a mirror server. See “DB_EXTENDED_PROPERTY function [System]” [SQL Anywhere Server - SQL Reference].

**New HTTP_RESPONSE_HEADER function** Returns the value of an HTTP response header. See “HTTP_RESPONSE_HEADER function [Web service]” [SQL Anywhere Server - SQL Reference].

**Enhancements to HTTP_VARIABLE function** You can now use the @BINARY attribute to return an x-www-form-urlencoded binary data value. See “HTTP_VARIABLE function [Web service]” [SQL Anywhere Server - SQL Reference].

**New ISENCRYPTED function** Determines if a string is encrypted. See “ISENCRIPTED function [System]” [SQL Anywhere Server - SQL Reference].

**New NEXT_HTTP_RESPONSE_HEADER function** Gets the next HTTP response header name. See “NEXT_HTTP_RESPONSE_HEADER function [Web service]” [SQL Anywhere Server - SQL Reference].

**New SWITCHOFFSET function** Returns a TIMESTAMP WITH TIME ZONE value that is converted from its original time zone offset to the specified time zone offset. See “SWITCHOFFSET function [Date and time]” [SQL Anywhere Server - SQL Reference].

**New SYSDATETIMEOFFSET function** Returns the current date, time, and time zone offset of the database server. See “SYSDATETIMEOFFSET function [Date and time]” [SQL Anywhere Server - SQL Reference].

**New TODATETIMEOFFSET function** Converts a TIMESTAMP value to a TIME STAMP WITH TIME ZONE value using the specified time zone offset. See “TODATETIMEOFFSET function [Date and time]” [SQL Anywhere Server - SQL Reference].

**New COUNT_BIG function** Counts the number of rows in a group depending on the specified parameters. See “COUNT_BIG function [Aggregate]” [SQL Anywhere Server - SQL Reference].
SQL statements

Following is a list of SQL enhancements introduced in SQL Anywhere version 12.0.0.

- **Improved qualification for named index hints**  A PRIMARY KEY index and a FOREIGN KEY index on a table can have the same name. When this is true, a named index hint cannot be unambiguously specified. The specification of a named index hint has been extended to allow the qualification of a hinted index as a PRIMARY KEY index or a FOREIGN KEY index. See “INDEX clause, FROM clause” [SQL Anywhere Server - SQL Reference].

- **New IS DISTINCT FROM and IS NOT DISTINCT FROM search conditions**  The IS DISTINCT FROM and IS NOT DISTINCT FROM search conditions allow you to control the evaluation of equality in the case of NULLs. See “IS DISTINCT FROM and IS NOT DISTINCT FROM search conditions” [SQL Anywhere Server - SQL Reference].

- **CREATE SYNCHRONIZATION PROFILE statement**  This statement creates a SQL Anywhere synchronization profile. A synchronization profile defines how a SQL Anywhere database synchronizes with the MobiLink server. See “CREATE SYNCHRONIZATION PROFILE statement [MobiLink]” [SQL Anywhere Server - SQL Reference].

- **New WITH NULLS NOT DISTINCT clause, CREATE INDEX statement**  A new clause, WITH NULLS NOT DISTINCT, has been added to the CREATE INDEX statement for use when creating UNIQUE indexes. This clause allows you to specify that NULLs in index keys are not unique. You must upgrade or rebuild existing databases to use this feature. See the UNIQUE clause of the “CREATE INDEX statement” [SQL Anywhere Server - SQL Reference].

- **Back quote identifier delimiter**  Back quotes can now be used as an identifier delimiter. See “Identifiers” [SQL Anywhere Server - SQL Reference].

- **New SAVE OPTION VALUES clause for ALTER TEXT CONFIGURATION statement**  Use this new clause to change the values of the date_format, time_format, timestamp_format, and timestamp_with_time_zone_format options saved with a text configuration object. See “ALTER TEXT CONFIGURATION statement” [SQL Anywhere Server - SQL Reference].

- **ALTER SERVER and CREATE SERVER statements**  The new IQODBC and IQJDBC server classes allow you to specify an SAP Sybase IQ server as a remote connection. See “ALTER SERVER statement” [SQL Anywhere Server - SQL Reference] and “CREATE SERVER statement” [SQL Anywhere Server - SQL Reference].

- **New LOCK FEATURE statement**  This statement prevents other concurrent connections from using a database server feature. See “LOCK FEATURE statement” [SQL Anywhere Server - SQL Reference].

- **Enhancements to the BEGIN, CREATE VARIABLE, and DECLARE statements**  Variable declarations can now include an initial starting value for the variable. See:
  - “BEGIN statement” [SQL Anywhere Server - SQL Reference]
  - “CREATE VARIABLE statement” [SQL Anywhere Server - SQL Reference]
  - “DECLARE statement” [SQL Anywhere Server - SQL Reference]
• **Enhancements to the LOAD TABLE statement**  The LOAD TABLE statement now supports the load-option clause, which allows you to control how data is loaded. It also supports the value XML in the FORMAT clause. See “LOAD TABLE statement” [SQL Anywhere Server - SQL Reference].

• **New IF NOT EXISTS clause**  With the new IF NOT EXISTS clause, no changes are made and an error is not returned if the named object already exists. See:
  - “CREATE INDEX statement” [SQL Anywhere Server - SQL Reference]
  - “CREATE PUBLICATION statement [MobiLink] [SQL Remote]” [SQL Anywhere Server - SQL Reference]
  - “CREATE SPATIAL REFERENCE SYSTEM statement” [SQL Anywhere Server - SQL Reference]
  - “CREATE TABLE statement” [SQL Anywhere Server - SQL Reference]
  - “CREATE TEXT INDEX statement” [SQL Anywhere Server - SQL Reference]

• **New IF EXISTS clause**  The new IF EXISTS clause allows you to specify that you do not want an error returned when the DROP statement attempts to remove a database object that does not exist. See:
  - “DROP EVENT statement” [SQL Anywhere Server - SQL Reference]
  - “DROP FUNCTION statement” [SQL Anywhere Server - SQL Reference]
  - “DROP INDEX statement” [SQL Anywhere Server - SQL Reference]
  - “DROP MATERIALIZED VIEW statement” [SQL Anywhere Server - SQL Reference]
  - “DROP PROCEDURE statement” [SQL Anywhere Server - SQL Reference]
  - “DROP PUBLICATION statement [MobiLink] [SQL Remote]” [SQL Anywhere Server - SQL Reference]
  - “DROP SPATIAL REFERENCE SYSTEM statement” [SQL Anywhere Server - SQL Reference]
  - “DROP SPATIAL UNIT OF MEASURE statement” [SQL Anywhere Server - SQL Reference]
  - “DROP SYNCHRONIZATION PROFILE statement [MobiLink]” [SQL Anywhere Server - SQL Reference]
  - “DROP TABLE statement” [SQL Anywhere Server - SQL Reference]
  - “DROP TRIGGER statement” [SQL Anywhere Server - SQL Reference]
  - “DROP VARIABLE statement” [SQL Anywhere Server - SQL Reference]
  - “DROP VIEW statement” [SQL Anywhere Server - SQL Reference]

• **CASE statement enhancements**  CASE statements are now supported in Transact-SQL procedures and batches.
• **New OR REPLACE clause**  The new OR REPLACE clause allows you to create or replace a profile or variable of the same name. See:

  ○ “CREATE FUNCTION statement [External call]” [SQL Anywhere Server - SQL Reference]
  ○ “CREATE FUNCTION statement [Web service]” [SQL Anywhere Server - SQL Reference]
  ○ “CREATE FUNCTION statement” [SQL Anywhere Server - SQL Reference]
  ○ “CREATE MIRROR SERVER statement” [SQL Anywhere Server - SQL Reference]
  ○ “CREATE PROCEDURE statement [External call]” [SQL Anywhere Server - SQL Reference]
  ○ “CREATE PROCEDURE statement [Web service]” [SQL Anywhere Server - SQL Reference]
  ○ “CREATE PROCEDURE statement [T-SQL]” [SQL Anywhere Server - SQL Reference]
  ○ “CREATE PROCEDURE statement” [SQL Anywhere Server - SQL Reference]
  ○ “CREATE SEQUENCE statement” [SQL Anywhere Server - SQL Reference]
  ○ “CREATE SPATIAL REFERENCE SYSTEM statement” [SQL Anywhere Server - SQL Reference]
  ○ “CREATE SYNONYMS FOR statement” [SQL Anywhere Server - SQL Reference]
  ○ “CREATE TRIGGER statement” [SQL Anywhere Server - SQL Reference]
  ○ “CREATE VIEW statement” [SQL Anywhere Server - SQL Reference]

• **New LIMIT clause support for SELECT statements**  You can now specify row counts and offsets in a SELECT statement using the new LIMIT clause. See “SELECT statement” [SQL Anywhere Server - SQL Reference].

• **Enhancement to the SET OPTION statement**  Syntax 1 of the SET OPTION statement now supports setting an option using the contents of a variable. See “SET OPTION statement” [SQL Anywhere Server - SQL Reference].

• **New IF [ NOT] OF search condition**  The IF [NOT] OF type-expression search condition has been added. See “Search conditions” [SQL Anywhere Server - SQL Reference] and “Spatial data type syntax” [SQL Anywhere Server - Spatial Data Support].

• **INSERT statement enhancements**  The following enhancements have been made to the INSERT statement. See “INSERT statement” [SQL Anywhere Server - SQL Reference].

  ○ **Support for more than one list of values**  An INSERT statement can now contain more than one list of values, allowing several rows to be inserted at once. For example:

  ```sql
  INSERT INTO T (c1,c2,c3)
  VALUES (1,10,100), (2,20,200), (3,30,300);
  ```

  ○ **Support for inserting rows with all default values**  SQL Anywhere allows the VALUES clause to contain specified values for a subset of the columns in the table. All unspecified columns are given default values as specified for each column by means of DEFAULT, NULL and COMPUTE clauses of the CREATE TABLE statements. Previously, the database server required that you specify input values for at least one of the columns in the table.

  Now, however, all columns can be given their default values by using either of the following syntax extensions:

  ```sql
  INSERT [ INTO ] table-name options DEFAULT VALUES ...
  ```
**SQL Anywhere new features**

**INSERT** [ INTO ] *table-name* ( ) *options* VALUES ( ) [ , ( ) ... ]

Specifying DEFAULT VALUES or VALUES is semantically equivalent to using the following syntax, where the number of default entries is equal to the number of columns in the table:

**INSERT** [ INTO ] *table-name* VALUES( default, default, ..., default )

The DEFAULT VALUES clause is part of the SQL/2008 standard, whereas the VALUES clause is a vendor extension.

- **New START SERVER statement**  The START SERVER statement has been added. It should be used instead of the START ENGINE statement, which is now deprecated. See “START SERVER statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference].

- **New STOP SERVER statement**  The STOP SERVER statement has been added. It should be used instead of the STOP ENGINE statement, which is now deprecated. See “STOP SERVER statement” [SQL Anywhere Server - SQL Reference].

- **New CHECKSUM clause for the ALTER DATABASE statement**  The CHECKSUM OFF clause lets you disable global checksums for databases. Global checksums are enabled by default for new version 12 databases. See “ALTER DATABASE statement” [SQL Anywhere Server - SQL Reference].

- **DELETE statement enhancement**  The DELETE statement now supports correlation names. See “DELETE statement” [SQL Anywhere Server - SQL Reference].

**Data types**

Following is a list of enhancements to data types introduced in SQL Anywhere version 12.0.0.

- **TIMESTAMP WITH TIME ZONE data type**  Stores a point in time with a time zone offset. See “TIMESTAMP WITH TIME ZONE data type” [SQL Anywhere Server - SQL Reference].

  The alias DATETIMEOFFSET is also supported. See “DATETIMEOFFSET data type” [SQL Anywhere Server - SQL Reference].

- **CHAR, NCHAR,NVARCHAR, and VARCHAR data types support 32767 character length**  The NCHAR and NVARCHAR data types can now be declared up to 32767 characters. The CHAR and VARCHAR data types with character-length semantics can now be declared up to 32767 characters. If the described byte length in the client character set for the NCHAR, NVARCHAR, CHAR, or VARCHAR data types with character length semantics would be more than 32767, then these types are described as the LONG NVARCHAR or LONG VARCHAR data type. The described byte length for CHAR and VARCHAR data types with byte length semantics now accounts for the maximum possible character set expansion when converting values to the client character set. If the resulting length including the maximum possible expansion is greater than 32767, the type is
described as LONG VARCHAR. There is no expansion if the client is using a single byte character set, or the client character set and the database character set encoding are the same. See:

- “CHAR data type” [SQL Anywhere Server - SQL Reference]
- “NCHAR data type” [SQL Anywhere Server - SQL Reference]
- “NVARCHAR data type” [SQL Anywhere Server - SQL Reference]
- “VARCHAR data type” [SQL Anywhere Server - SQL Reference]

- **Data types for spatial data**  
  Several new data types have been added to support spatial data. See “Accessing and manipulating spatial data” [SQL Anywhere Server - Spatial Data Support].

**Programming interfaces**

Following is a list of enhancements to programming interfaces introduced in SQL Anywhere version 12.0.0.

- **Web services performance enhancements**  
  HTTP services now support automatic connection pooling to maximize the effect of plan caching and benefit from the potential performance improvement. See “How to develop web service applications in an HTTP web server” [SQL Anywhere Server - Programming].

- **Support for customized ODBC driver names**  
  To facilitate the installation and registration of multiple independent copies of the SQL Anywhere ODBC driver on a client system, you can now assign a customized name to the ODBC driver. See “ODBC driver configuration” [SQL Anywhere Server - Programming].

- **ANSI-only ODBC driver for Unix**  
  Versions of Unix ODBC driver managers that define SQLWCHAR as 32-bit (UTF-32) quantities cannot be used with the SQL Anywhere ODBC driver that supports wide calls since this driver is built for 16-bit SQLWCHAR. For these cases, an ANSI-only version of the SQL Anywhere ODBC driver is provided. This version of the ODBC driver does not support the wide call interface (SQLConnectW for example). See “UTF-32 ODBC driver managers for Unix” [SQL Anywhere Server - Programming].

- **DBTools: no_reload_status bit field added to an_unload_db structure**  
  A new bit field, no_reload_status, has been added to the an_unload_db structure. You can use no_reload_status to suppress table and index progress messages on reload. See “an_unload_db structure [database tools]” [SQL Anywhere Server - Programming].

- **New SQL Anywhere TYPE-2 JDBC driver**  
  A new TYPE-2 JDBC driver is now available for JDBC applications to use when connecting to SQL Anywhere. Unlike the iAnywhere JDBC driver, which sits on top of ODBC and can be used to connect to a variety of servers via ODBC, the SQL Anywhere JDBC driver connects to SQL Anywhere only and does not require the SQL Anywhere ODBC driver to be installed or registered.

The SQL Anywhere JDBC driver comes with a JDBC 3.0 driver and a JDBC 4.0 driver.

**Note**

If you currently use the iAnywhere JDBC driver, it is strongly recommended that you change to the new SQL Anywhere JDBC driver.
The JDBC 4.0 driver automatically loads and registers itself.

To use the version 3.0 SQL Anywhere JDBC driver, you must load the sybase.jdbc.sqlanywhere.IDriver class that implements the java.sql.Driver interface and registers the SQL Anywhere JDBC driver with the JDBC driver manager. Once loaded, connections using the SQL Anywhere JDBC driver can be made by using the URL jdbc:sqlanywhere:connection-string-parameters. The connection-string-parameters are the standard connection parameters required to connect to SQL Anywhere. Note that the application no longer needs to specify DRIVER= or DSN= in connection-string-parameters when using the SQL Anywhere JDBC driver. See “JDBC support” [SQL Anywhere Server - Programming].

- **New support for JDBC statement class methods** Previously, the JDBC drivers only supported the addBatch and executeBatch methods of the PreparedStatement class. The JDBC drivers now also support the addBatch, clearBatch, and executeBatch methods of the Statement class. Due to the fact that the JDBC specification is unclear on the behavior of the executeBatch method of the Statement class, the following notes should be considered when using this method with the SQL Anywhere JDBC drivers:

  1. Processing of the batch stops immediately upon encountering a SQL exception or result set. If processing of the batch stops, then a BatchUpdateException will be thrown by the executeBatch method. Calling the getUpdateCounts method on the BatchUpdateException will return an integer array of row counts where the set of counts prior to the batch failure will contain a valid non-negative update count; while all counts at the point of the batch failure and beyond will contain a -1 value. Casting the BatchUpdateCount to a SQLException provides additional details as to why batch processing was stopped.

  2. The batch is only cleared when the clearBatch method is explicitly called. As a result, calling the executeBatch method repeatedly will re-execute the batch over and over again. In addition, calling execute( sql_query ) or executeQuery( sql_query ) correctly executes the specified SQL query, but does not clear the underlying batch. Hence, calling the executeBatch method followed by execute( sql_query ) followed by the executeBatch method again will execute the set of batched statements, then execute the specified SQL query, and then execute the set of batched statements again.

See “JDBC support” [SQL Anywhere Server - Programming].

- **RESUME statement returns row counts or the row estimate** The RESUME statement now returns the number of rows in the result set or the row estimate for the next result set in a procedure call. In previous versions, this row count was not available after the RESUME statement. To get the RESUME row count, both the client application and the database server must be SQL Anywhere 12.

This change affects the following client APIs:

<table>
<thead>
<tr>
<th>API</th>
<th>Function call or statement affected</th>
<th>Row count returned by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded SQL</td>
<td>RESUME statement</td>
<td>SQLCOUNT field</td>
</tr>
<tr>
<td>ODBC</td>
<td>SQLMoreResults function</td>
<td>SQLRowCount function</td>
</tr>
</tbody>
</table>
### Catalog changes

Following is a list of catalog changes introduced in SQL Anywhere version 12.0.0.

- **New ISYSSEQUENCE system table and SYSSEQUENCE system view**: The ISYSSEQUENCE system table contains one row for each user-defined sequence. See “SYSSEQUENCE system view” [SQL Anywhere Server - SQL Reference].
• **New ISYSSEQUENCEPERM system table and SYSSEQUENCEPERM system view**  The ISYSSEQUENCEPERM system table records the privileges that users or groups hold on sequences. See “SYSSEQUENCEPERM system view” [SQL Anywhere Server - SQL Reference].

• **ISYSTEXTCONFIG system table**  The ISYSTEXTCONFIG system table has been modified to hold information about the entry points and the external libraries used for tokenizing and/or prefiltering. Specifically, the existing prefilter column data type has changed to be a LONG VARCHAR to hold the entry points and library name for an external prefilter library. A new LONG VARCHAR column, external_term_breaker, has been added to hold the entry points and library name for an external term breaker library. See “SYSTEXTCONFIG system view” [SQL Anywhere Server - SQL Reference].

• **ISYSTABCOL system table: new base_type_str column**  This column holds the annotated type string representing the physical type of the column.

• **ISYSPROCPARM system table: new base_type_str column**  This column holds the annotated type string representing the physical type of the parameter.

• **ISYSUSERTYPE system table: new base_type_str column**  This column holds the annotated type string representing the physical type of the user type.

• **ISYSOBJECT system table: new object_type_str column**  This column holds a word description (for example, MAT VIEW) of the value in SYSOBJECT.object_type.

  Previously, the SYSOBJECT view only provided object_type, which gave the object type expressed as a TINYINT. This meant you needed to access either the documentation or the view definition to translate the integer to a word description. Now, you can query the object_type_str column to see the word description of the object.

---

**Windows Mobile enhancements**

Following is a list of Windows Mobile enhancements introduced in SQL Anywhere version 12.0.0.

• **Default stack size for internal execution threads changed for Windows Mobile**  The default stack size on Windows Mobile is now 96 KB, the minimum stack size is now 64 KB, and the maximum stack size is now 512 KB. See “-gss database server option” [SQL Anywhere Server - Database Administration].

---

**Unix/Linux enhancements**

Following is a list of Unix and Linux enhancements introduced in SQL Anywhere version 12.0.0.

• **Performance Statistics utility (dbstats)**  The dbstats utility returns the value of performance statistics on Unix computers. Its behavior is similar to that of the Windows Performance Monitor. See “Performance Statistics utility (dbstats) (Unix)” [SQL Anywhere Server - Database Administration].
- **PID file created for Linux services**  When a SQL Anywhere service is running on Linux, a PID file is created in the /var/run directory. See “Service utility (dbsvc) for Linux” [SQL Anywhere Server - Database Administration].

### Performance enhancements

Following is a list of performance enhancements introduced in version 12.0.0 for which there are no user-visible changes other than performance improvement.

- **Improvements to locking when updating primary rows**  Updating a non-key column in a primary row and modifying foreign rows referring to that row no longer interfere with each other. For example, in the sample database, a sales_order row can be added while the corresponding customer address is being updated, without one operation having to wait on the other. See “sa_locks system procedure” [SQL Anywhere Server - SQL Reference] and “Types of locks” [SQL Anywhere Server - SQL Usage].

- **Reduced locking at isolation levels 2 and 3**  Read locks on individual rows are no longer obtained when a table is locked in share mode (LOCK TABLE...IN SHARE MODE statement). This can result in reduced locking overhead at isolation levels 2 and 3.

- **Improved index performance**  SQL Anywhere 12 includes enhanced algorithms and a new on-disk layout to improve the performance of deleting large numbers of clustered sequential values from an index.

- **Improved validation performance**  SQL Anywhere 12 includes many enhancements to improve the validation of large databases.

- **Improved request prioritization**  SQL Anywhere 12 has been enhanced to boost the priority of I/O bound requests, which results in better utilization of hardware resources.

- **Improved remote data access performance**  SQL Anywhere 12 includes many enhancements to improve remote data access performance, including improved proxy table performance.

- **New cost model**  SQL Anywhere 12 includes a CPU cost model that more accurately estimates query execution costs on modern hardware. This behavior may cause changes to access plans for some queries.

- **Enhancement to queries embedded in user-defined functions**  SQL queries embedded in user-defined functions can now be in-lined by the query optimizer, which avoids a procedure context switch with each invocation and provides the optimizer with new degrees of freedom with which to optimize the statement.

- **Improvements when converting expressions to different data types**  Improvements have been made to the evaluation rules the database server uses when converting an expression to a different data type. The new evaluation rules make the conversion more efficient to execute.
**Miscellaneous**

Following is a list of miscellaneous enhancements introduced in SQL Anywhere version 12.0.0.

- **New script for making copies of the sample database**  
  The `newdemo.bat` and `newdemo.sh` files are located in the `bin32` or `bin64` directory of your SQL Anywhere installation directory and can be used to create a copy of the sample database that includes all the data from the sample database. This script can be used to re-create the sample database or to make new copies of it with a different name. See [Recreate the sample database (demo.db)](SQL Anywhere 16 - Introduction).

- **New selectivity estimate source type for the query optimizer**  
  A new selectivity estimate source type, JOIN, has been added. This new source type is used by the query optimizer for selectivity estimations of atomic predicates of the form `T.X = R.X`. See [“ESTIMATE_SOURCE function”](SQL Anywhere Server - SQL Reference).

- **Better handling of overflow errors**  
  Arithmetic operations (`+`, `-`, `*`, `/`, `SUM`, `AVG`) may overflow because the result of the operation cannot be represented in the data type. Previously, for expressions of type `INT`, this overflow returned an error, while for all other data types, the overflow resulted in an undefined value. Now, all arithmetic operations on all types detect overflow and return an error if the result cannot be expressed in the data type.

- **Use the ALTER EXTERNAL ENVIRONMENT statement to set the location of dbmlsync**  
  If, during synchronization, the `dbmlsync` executable cannot be located using the PATH environment variable that the database server is using, you can now use the ALTER EXTERNAL ENVIRONMENT statement to tell the database server the location of the `dbmlsync` executable. See [“ALTER EXTERNAL ENVIRONMENT statement”](SQL Anywhere Server - SQL Reference).

- **Back quotes are supported as delimiters**  
  You can now use back quotes (``) to delimit identifiers in SQL Anywhere. See [“Identifiers”](SQL Anywhere Server - SQL Reference).

- **Japanese Unicode Collation Algorithm (UCA) collation tailoring option**  
  A new Japanese UCA collation tailoring option is now available. You can use it to define a primary-level difference between all Hiragana and Katakana letters. This new tailoring option provides correct equality comparisons of Hiragana and Katakana letters in case-insensitive collations. See [“Collation tailoring options”](SQL Anywhere Server - Database Administration).

- **Changes to the server messages window and Windows system tray icon**  
  The title bar for the database server messages window now specifies whether you are running a personal server or a network server. The tooltip for the Windows system tray icon also specifies the type of database server. As well, the database server About window includes the edition of SQL Anywhere that is running.

- **Information utility (dbinfo) enhancement**  
  The `dbinfo` utility now returns information about the CHAR collation specification, the CHAR encoding, the NCHAR collation specification, or the NCHAR encoding for a database. See [“Information utility (dbinfo)”](SQL Anywhere Server - Database Administration).

- **Controlling the amount of address space reserved for non-cache use**  
  The `-ch` option now leaves more address space for use outside the cache, and the maximum non-AWE cache size on 32-bit operating systems has been reduced. See [“-ch database server option”](SQL Anywhere Server - Administrator’s Guide).
Enhance priority management for I/O bound compared to CPU bound requests  The database server now dynamically detects requests that are I/O bound and increases their priorities over CPU-bound tasks to increase disk throughput and the use of hardware resources.

Improved robustness across power failures on Windows  You can set Windows Registry entries to improve robustness across power failures on systems using certain Intel storage drivers when deploying SQL Anywhere. Failure to set this parameter can result in lost data and corrupted databases in the event of a power failure. To determine if these entries are required as part of your deployment, see “Improving robustness on Intel storage drivers” [SQL Anywhere Server - Programming].

SQL Flagger enhancement  The SQL Flagger now supports the SQL/2008 standard. See “SQL compliance testing using the SQL Flagger” [SQL Anywhere Server - SQL Usage].

Progress messages  Some SQL statements now support sending progress messages from the database server to the client. See “New progress_messages option” on page 125.

SQL Anywhere behavior changes

Following is a list of behavior changes to SQL Anywhere introduced in version 12.0.0. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

Checksums enabled by default for new databases  When you create a new database, global checksums are now enabled by default. Global checksums are calculated and verified each time a database page is read or written, and are used to determine whether a database page has been modified on disk. You can control whether global checksums are enabled for a database by using the -s[ + | - ] option for the Initialization utility, or by using the CHECKSUM clause of the CREATE DATABASE statement. See:

- “Checksum enhancements” on page 122
- “Initialization utility (dbinit)” [SQL Anywhere Server - Database Administration]
- “CREATE DATABASE statement” [SQL Anywhere Server - SQL Reference]
- “START DATABASE statement” [SQL Anywhere Server - SQL Reference]

Checkpoint log changes  In previous releases, when you stopped a database, SQL Anywhere completely truncated the checkpoint log. In version 12, a history of the checkpoint log usage is maintained in the database and is used to determine an appropriate size for the checkpoint log for the next session.

Maintaining the checkpoint log across sessions avoids the overhead of allocating it the next time the database is started and avoids file fragmentation that can occur when files are extended. Database files will now be larger when the database is shut down than they were in previous releases, but the additional space is reused for the checkpoint log the next time the database is restarted. See “Checkpoint logs” [SQL Anywhere Server - Database Administration].
• **Network database server now allocates the maximum number of workers**  
In previous releases, at startup the network database server allocated the number of workers that corresponded to the database server multiprogramming level. In version 12 network servers, the number of workers allocated corresponds to the maximum server multiprogramming level. This increase in the number of workers increases the address space requirements of the network database server for worker stacks, and may impact the amount of database cache that the database server can allocate.

For example, on 32-bit Windows platforms, by default each worker requires 1 MB of address space for its stack. A version 11 network server starting on Windows would require 20 MB of address space for worker stacks, as its default multiprogramming level is 20. However, a version 12 network server starting on Windows requires 80 MB of address space, as the default maximum number of workers is 80. This change does not affect personal servers or Windows Mobile. See “SQL Anywhere threading” [SQL Anywhere Server - Database Administration].

• **Old statistics are not loaded when a database is rebuilt**  
When you rebuild a version 11 or earlier database, the LOAD STATISTICS statement silently skips loading old string statistics into the new database, but upgrades the version of the string statistics. See “LOAD STATISTICS statement” [SQL Anywhere Server - SQL Reference].

Upgrading a database (using the Upgrade utility) does not upgrade the version of the string statistics.

• **Positioned DELETE and UPDATE statements**  
Previously, you could specify a TOP or FIRST clause in a positioned UPDATE or DELETE statement; however, the clauses would be ignored. Now, specifying TOP or FIRST in a positioned UPDATE or DELETE statement returns a syntax error. See “UPDATE (positioned) statement [ESQL] [SP]” [SQL Anywhere Server - SQL Reference], and “DELETE statement (positioned) [ESQL] [SP]” [SQL Anywhere Server - SQL Reference].

• **JDBC drivers now report CHAR and VARCHAR instead of just CHAR for both**  
Previously, when an application connected using the iAnywhere JDBC driver and attempted to describe the metadata of a table or result set that contains a CHAR column, the metadata would return the type name of the column as CHAR but the SQL type would still come back as Types.VARCHAR. If a JDBC application wanted to get the JDBC driver to return the SQL type of CHAR columns as Types.CHAR, then the application was required to set the odbc_distinguish_char_and_varchar database option. Now, the new SQL Anywhere JDBC driver and the deprecated iAnywhere JDBC driver return the type name CHAR and SQL type Types.CHAR for table and result set columns of type CHAR, and the type name VARCHAR and the SQL type Types.VARCHAR for columns of type VARCHAR regardless of the database option setting.

• **Changes to CURRENT UTC TIMESTAMP and UTC TIMESTAMP special values**  
The underlying data type for the CURRENT UTC TIMESTAMP special value, and the default value UTC TIMESTAMP special value, is now TIMESTAMP WITH TIME ZONE. If these values are used with columns defined as TIMESTAMP, the time zone offset will be truncated and no difference in behavior should be noticeable. However, if these values are used with CHAR or VARCHAR columns, the offset will result in different values being generated than before. See “CURRENT UTC TIMESTAMP special value” [SQL Anywhere Server - SQL Reference] and “UTC TIMESTAMP special value” [SQL Anywhere Server - SQL Reference].

• **Personal server no longer starts TCP/IP by default**  
The personal database server only starts the shared memory protocol by default. If you want to use the TCP/IP protocol, you must specify it
using the -x server option when starting the personal database server. See “-x database server option” [SQL Anywhere Server - Database Administration] and “Communication protocol considerations” [SQL Anywhere Server - Database Administration].

- **TCP/IP connections** If you are connecting over TCP/IP, the database server name (specified by the ServerName (SERVER) connection parameter) is no longer mandatory if a host name is provided with the HOST connection parameter. See “Host connection parameter” [SQL Anywhere Server - Database Administration].

- **Host (IP) protocol option** In previous releases, the Host protocol option indicated one or more hosts on which the database server may be running, but was considered a hint to the client library. If the database server was not found on those hosts, a network broadcast was done to find the database server. In this release, if the Host option is specified, only the specified hosts are searched for the database server and the client does not broadcast to find the database server by default.

  This behavior is equivalent to setting the DoBroadcast protocol option to Direct. If the database server is running on a computer other than the ones specified by the HOST protocol option, it is not found. If you want the same behavior as previous releases, specify DoBroadcast=All in the connection string. See “Host (IP) protocol option (client side only)” [SQL Anywhere Server - Database Administration] and “DoBroadcast (DOBROAD) protocol option” [SQL Anywhere Server - Database Administration].

- **ServerPort (PORT) protocol option** In previous releases, the PORT protocol option indicated one or more port numbers on which the database server may be listening, but was considered a hint to the client library. When the client library sent out a broadcast, it would use the port numbers specified by the PORT protocol option, as well as the default port number, 2638. In this release, if the PORT option is specified, only the specified ports are used to find the database server. See “ServerPort (PORT) protocol option” [SQL Anywhere Server - Database Administration].

- **Idle connection parameter respected for shared memory connections** The Idle connection parameter specifies a connection's idle timeout period. Shared memory connections now respect this connection parameter. The default idle timeout for shared memory connections is zero (never idle timeout). See “Idle connection parameter” [SQL Anywhere Server - Database Administration].

- **Database server name may not be the same as the name specified by the ServerName (Server) connection parameter** In previous releases, the value of the Name database property always matched the value specified in the ServerName (Server) connection parameter. However, if a client connects to a database and uses the new NodeType (NODE) connection parameter and the client is redirected to connect to a different database server, then the names do not match. See “NodeType (NODE) connection parameter” [SQL Anywhere Server - Database Administration].

- **Some operations are not supported when connecting with an alternate server name** In previous releases, if a client connected to a database using an alternate server name, they could create, stop, and drop other databases on the same database server. These operations are no longer supported when you connect using an alternate server name.

- **Database mirroring behavior changes and deprecated features** Using the -xp server option to define database mirroring options such as the name of the arbiter server, the authentication string, and the synchronization mode is deprecated. However, you must still specify -xp on if you want to use the database server in a mirroring system. See “-xp database option” [SQL Anywhere Server - Database Administration].
You can now define the database mirroring settings using the following SQL statements:

- “CREATE MIRROR SERVER statement” [SQL Anywhere Server - SQL Reference]
- “ALTER MIRROR SERVER statement” [SQL Anywhere Server - SQL Reference]
- “SET MIRROR OPTION statement” [SQL Anywhere Server - SQL Reference]

The following behavior changes to database mirroring have been introduced in this release:

- In previous releases, the name of the state information file for a mirror server defaulted to a name based on the server name. You must now specify the name of the state information file.
- In previous releases, web service requests directed to the mirror server were redirected to the primary server. In this release, web service requests are handled by the server that receives them.

For information about the enhancements to database mirroring in this release, see “Database mirroring enhancements” on page 118.

- **Embedded SQL cursor behavior change** Embedded SQL cursors now default to READ ONLY. Explicit FOR READ ONLY or FOR UPDATE clauses must now be specified in the PREPARE statement and not the DECLARE statement. See:
  - “DECLARE CURSOR statement [ESQL] [SP]” [SQL Anywhere Server - SQL Reference]
  - “PREPARE statement [ESQL]” [SQL Anywhere Server - SQL Reference]

- **Cursors closed for CREATE OR REPLACE PROCEDURE statements** When you execute a CREATE OR REPLACE PROCEDURE statement, any cursors that are open for a connection are closed. See “CREATE PROCEDURE statement” [SQL Anywhere Server - SQL Reference].

- **Back quote identifier delimiter** Back quotes can now be used as an identifier delimiter. See “Identifiers” [SQL Anywhere Server - SQL Reference].

- **Changes in locking behavior** To maximize concurrency, the key and non-key portions of a row can now be locked independently. Non-key columns of a row can be updated without interfering with the insertion and deletion of foreign rows referencing that row. See “How locking works” [SQL Anywhere Server - SQL Usage].

- **ODBC driver behavior change** The ODBC function SQLTables can be used to get a list of all schemas (users) by calling the function with the SQL_ALL_SCHEMAS argument. In previous versions, the list of users returned by this function only included users that owned a table. In a newly initialized database, this excluded some users, and in particular the DBA user. Now the complete list of schemas is returned, including those that do not own a table.

- **divide_by_zero_error option** The divide_by_zero_error option was not supported in version 11. This option is supported in version 12. If you are using materialized views, you must set this option to On (the default) to create new materialized views. See “divide_by_zero_error option” [SQL Anywhere Server - Database Administration] and “Materialized views restrictions” [SQL Anywhere Server - SQL Usage].

- **PrefetchBuffer (PBUF) connection parameter** In previous releases, the amount of memory for buffering rows specified by the PrefetchBuffer (PBUF) connection parameter was shared between all connections. In this release, the amount of memory specified by this connection parameter is available
to each connection. See “PrefetchBuffer (PBUF) connection parameter” [SQL Anywhere Server - Database Administration].

- **External logins automatically removed for dropped users**  When you remove a user from the database, any external logins for the user are now dropped automatically. In previous releases, you had to remove the external logins separately. See “Deleting a user (Sybase Central)” [SQL Anywhere Server - Database Administration].

- **The database cleaner and database validation can no longer operate at the same time**  In previous releases, database validation and the database cleaner could run at the same time, and report errors because of concurrent access to database pages. Database validation and the database cleaner no longer simultaneously operate on the same database. Validation waits for the database cleaner to finish, and the database cleaner waits for validation to finish if the database cleaner is started by calling sa_clean_database. The database cleaner can operate simultaneously with table or index validation.

- **Data types changed for columns in sa_index_density system procedure**  The data type of the density and skew columns has been changed from numeric(8,6) to double. See “sa_index_density system procedure” [SQL Anywhere Server - SQL Reference].

- **DATEDIFF function**  In previous releases, the DATEDIFF function returned an INTEGER for date parts of hours and smaller. DATEDIFF now returns an a BIGINT for these date parts. See “DATEDIFF function [Date and time]” [SQL Anywhere Server - SQL Reference].

- **OPENXML operator**  The OPENXML string operator is no longer owned by the dbo database user. Any queries that use the OPENXML operator and qualify the name with dbo, must be changed to remove dbo. Running a statement similar to the following now returns an error:

  ```sql
  SELECT ... FROM dbo.OPENXML(...)
  ```

  In previous releases, the OPENXML operator converted NCHAR data to the CHAR encoding and parsed the data in CHAR format. Now NCHAR data is parsed in the NCHAR encoding if there are NCHAR columns in the output.

  In previous releases, the XPath arguments in the WITH clause could only be literal strings. Now literal strings and variables are allowed. See “OPENXML operator” [SQL Anywhere Server - SQL Reference].

  Databases that are upgraded from version 11 or earlier contain a row for OPENXML in the SYS.SYSPROCEDURE system view, but that definition cannot be used in the version 12 database. A syntax error is returned if you attempt to use the operator as follows:

  ```sql
  SELECT * FROM dbo."OPENXML"(...)```

- **sa_text_index_vocab system procedure**  Attempting to call sa_text_index_vocab on an NCHAR text index now returns an error. Use the new sa_text_index_vocab_nchar system procedure instead. See “sa_text_index_vocab_nchar system procedure” [SQL Anywhere Server - SQL Reference].

  Additionally:
○ The `tab_owner` parameter is now optional.

○ `sa_text_index_vocab` can now be used with a `CALL` statement.

○ `sa_text_index_vocab` can now be used in a statement within a procedure.

○ The parameter values can now be host variables or expressions.

● **Default collation changed for Mac OS X**  In previous releases, in the absence of any environment variables on Mac OS X, the Initialization utility (dbinit) would use ISO_8859-1:1987 for its default CHAR character set and ISO1LATIN1 for its CHAR collation (the same behavior as Linux). It now chooses UTF-8 with the UTF8BIN collation. See “Recommended character sets and collations” [SQL Anywhere Server - Database Administration].

● **Reserved words**  The following is a list of reserved words added to the database in SQL Anywhere version 12.0.0:

  ○ `datetimeoffset`
  ○ `inner`
  ○ `openxml`
  ○ `spatial`
  ○ `treat`

The following is a list of reserved words removed from the database in SQL Anywhere version 12.0.0:

  ○ `index_lparen`
  ○ `lock`
  ○ `with_cube`
  ○ `with_lparen`
  ○ `syntax_error`
  ○ `with_rollup`

● **WITH (index-hint) clause behavior change**  Previously, if the primary key index, foreign key index, and/or the normal index on a table had the same name, the optimizer would resolve to the name of the primary or foreign key index. Now, the optimizer resolves to the name of the normal index. If one does not exist, the optimizer resolves to the name of the primary key first and then to the foreign key. See “WITH table-hint clause, FROM clause” [SQL Anywhere Server - SQL Reference].
SQL Anywhere deprecated and discontinued features

Note
As with all forward-looking statements, the lists of deprecated features are not guaranteed to be complete and are subject to change.

- **Address Windowing Extensions (AWE) deprecated**  The use of Address Windowing Extensions for 32-bit Windows is deprecated. If you need a large cache, it is recommended that you use the 64-bit version of the SQL Anywhere database server on a 64-bit operating system.

- **CALL statement**  Use of this statement to invoke a function is deprecated. If you have a function you want to call, consider using an assignment statement to invoke the function and assign its result to a variable. For example:

```
DECLARE varname INT; SET varname=test( );
```

See “CALL statement” [SQL Anywhere Server - SQL Reference].

- **STOP ENGINE statement**  The STOP ENGINE statement is deprecated. Use the STOP SERVER statement instead. See “STOP SERVER statement” [SQL Anywhere Server - SQL Reference].

- **Windows 2000 support removed**  As of version 12.0.0, SQL Anywhere is no longer supported on Windows 2000.

- **Rebuild utility removed**  The Rebuild utility is not supported in this release for rebuilding SQL Anywhere databases. You can rebuild databases using the Unload utility (dbunload). See “Unload utility (dbunload)” [SQL Anywhere Server - Database Administration].

- **Unsupported database properties**  The following properties have been removed in this release:
  - CheckpointLogBitmapPagesWritten database property
  - CheckpointLogBitmapSize database property
  - java_main_userid connection property
  - QueryRowsBufferFetch connection property
  - QueryRowsBufferFetch database property

- **JDBC-based server classes deprecated**  Support for the following JDBC-based server classes has been deprecated:
  - ASEJDBC
  - IQJDBC
  - SAJDBC

Applications should be updated to use the ODBC-based server classes. See “ODBC external server definitions” [SQL Anywhere Server - SQL Usage].
- **SQL Anywhere Explorer no longer supported**  The SQL Anywhere Explorer and SQL Anywhere Toolbar for Visual Studio are no longer supported. Use Microsoft's Server Explorer instead.

- **Short int embedded SQL indicator variable deprecated** To allow for the future use of 32- and 64-bit lengths and indicators, the use of short int for embedded SQL indicator variables is deprecated. Use a_sql_len instead. See “Indicator variables” [SQL Anywhere Server - Programming].

- **EngineName (ENG) connection parameter deprecated** The EngineName (ENG) connection parameter is deprecated. You can use the ServerName (Server) connection parameter instead. The short form of the ServerName connection parameter has been changed from ENG to Server. See “ServerName (Server) connection parameter” [SQL Anywhere Server - Database Administration].

- **iAnywhere JDBC driver deprecated** The Type 1 iAnywhere JDBC driver is deprecated. Use the Type 2 SQL Anywhere JDBC driver instead. See “New SQL Anywhere TYPE-2 JDBC driver” on page 134.

- **-gu all database server option deprecated** The value all for the -gu database server option is deprecated. See “-gu database server option” [SQL Anywhere Server - Database Administration].

- **-sm dbsrv12 database option (deprecated)** The -sm database option is deprecated. Use the CREATE MIRROR SERVER statement instead. See “CREATE MIRROR SERVER statement” [SQL Anywhere Server - SQL Reference].

- **SET OPTION statement** The ability to specify an identifier as an option value in a SET OPTION statement, rather than a string literal, has been deprecated.

- **Service utility (dbsvc utility)** The value Standalone for the -t option has been deprecated. Use -t Personal instead to create a service for the personal database server. See “Service utility (dbsvc) for Linux” [SQL Anywhere Server - Database Administration] and “Service utility (dbsvc) for Windows” [SQL Anywhere Server - Database Administration].

- **Host (IP) protocol option support from the server** Support for specifying the Host (IP) protocol option in a server command has been removed. For example, the following command is no longer supported and now returns an error: dbeng12 -x tcpip(host=host-name) "$SQLANY12\demo.db". However, you can still specify the Host (IP) protocol option in a CommLinks(LINKS) connection parameter from the client side. See “Host (IP) protocol option (client side only)” [SQL Anywhere Server - Database Administration].

**Note**
The Host protocol option is different from the Host connection parameter. The Host protocol option is used by the CommLinks connection parameter. You should only use the CommLinks (LINKS) connection parameter if you need to specify TCP/IP options other than Host and ServerPort (PORT). You cannot specify both CommLinks and Host in the connection string. See “CommLinks (LINKS) connection parameter” [SQL Anywhere Server - Database Administration].

In most cases, you should use the HOST connection parameter. See “Host connection parameter” [SQL Anywhere Server - Database Administration].
MobiLink new features

Following is a list of additions to MobiLink introduced in version 12.0.0. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

- **Central administration of remote databases**  
  Central administration of remote databases can be used to do the following:
  - Centrally control when a remote database synchronizes with MobiLink. This was previously set on the client with dbmlsync option or hard-coded into an application.
  - Push schema changes to remote databases.
  - Help diagnose problems with specific remote databases or with the synchronization system in general.

  For information about central administration of remote databases, see “Central administration of remote databases” [MobiLink - Server Administration].

- **New MobiLink Replay utility (mlreplay)**  
  The MobiLink Replay utility is a tool used to replay MobiLink protocol that is recorded by the MobiLink server. See “MobiLink Replay utility (mlreplay)” [MobiLink - Server Administration].

- **MobiLink 12 plug-in for Sybase Central has been redesigned**  
  The MobiLink plug-in has been redesigned in version 12 to support central administration of remote databases. The two MobiLink modes, Model and Admin, have been combined in the new plug-in. Now, you can use the MobiLink plug-in to create synchronization projects that contain consolidated databases, groups, synchronization models, and remote tasks. Old synchronization models can be imported into synchronization projects. See “Central administration of remote databases” [MobiLink - Server Administration].

- **Map consolidated columns to special values**  
  You can now map consolidated columns to special values (such as the ML user) instead of remote columns.

- **Download delete subsets now supported**  
  You can now specify download delete subsets, which by default are the same as the download subset.

- **New tree view allows selection of columns and tables**  
  You can now choose the columns as well as the tables when creating or updating a consolidated database, or choosing tables to be synchronized in an existing remote database. To facilitate this, there is now a handy tree view of tables and columns where you can place checkmarks next to the tables and columns you want to synchronize.

- **Support for spatial data types**  
  Synchronization models now support spatial data types and the TIMESTAMP WITH TIME ZONE data type.

- **Shadow tables now have indexes**  
  Indexes are now created for any shadow tables that are created by the synchronization model. This can speed up download_delete_cursor and download_cursor scripts that use shadow tables.
SQL Anywhere Monitor now supports MobiLink server farms and Relay Server farms
Now, you can use the SQL Anywhere Monitor to monitor MobiLink server farms and Relay Server farms as well as SQL Anywhere databases and MobiLink servers. See “SQL Anywhere Monitor” [SQL Anywhere Server - Database Administration].

New MobiLink arbiter server utility for server farms
For server-initiated synchronization the MobiLink arbiter ensures that only a single MobiLink server in a server farm is running as the primary server. See “Architecture” [MobiLink - Getting Started].

Enhanced support for server farms
New remote ID locking logic is used to prevent redundant synchronizations from the same remote ID in MobiLink high-availability. The -ss option no longer needs to be set. An arbiter is required when using server-initiated synchronization with a MobiLink server farm.

Note
Running the MobiLink server in a server farm is a feature of the MobiLink high availability option, which requires a separate license. See “Separately licensed components” [SQL Anywhere 16 - Introduction].

TLS cipher suite support has changed
MobiLink server and clients now support 256-bit AES cipher suites for both RSA and ECC. Also, support has been added for the RFC 4492 version of the ECC cipher suites.

Dynamic memory caching
With dynamic memory caching, there is increased use of the memory cache so a larger cache may be needed to prevent swapping. Overall memory use should still be about the same.

New integrated Outbound Enabler
Use the new OE protocol for the -x option for mlsrv12 to use an integrated Outbound Enabler instead of the stand-alone Outbound Enabler invoked with the rsoe command. See “-x mlsrv16 option” [MobiLink - Server Administration].

New Relay Server plug-in for Sybase Central
There is a new Relay Server plug-in for Sybase Central that enables you to configure back-end farms and servers. The Relay Server plug-in is only supported on Windows and Linux. See “Relay Server plug-in for Sybase Central” [Relay Server].

Upload and download data script requirement
Use the ml_add_missing_dnld_scripts stored procedure to fix missing download_cursor and/or download_delete_cursor scripts. See “ml_add_missing_dnld_scripts system procedure” [MobiLink - Server Administration].

SQL statements
Following is a list of SQL enhancements for MobiLink introduced in SQL Anywhere version 12.0.0.

New SYNCHRONIZE statement [MobiLink]
This new statement allows you to synchronize a SQL Anywhere remote database with a MobiLink server. See “SYNCHRONIZE statement [MobiLink]” [SQL Anywhere Server - SQL Reference].

New START SYNCHRONIZATION SCHEMA CHANGE statement [MobiLink]
This new statement allows you to start a MobiLink synchronization schema change in a SQL Anywhere remote
database. See “START SYNCHRONIZATION SCHEMA CHANGE statement [MobiLink]” [SQL
Anywhere Server - SQL Reference].

- **New STOP SYNCHRONIZATION SCHEMA CHANGE statement [MobiLink]** This new
  statement allows you to stop a MobiLink synchronization schema change in a SQL Anywhere remote
database. See “STOP SYNCHRONIZATION SCHEMA CHANGE statement [MobiLink]” [SQL
Anywhere Server - SQL Reference].

- **CREATE SYNCHRONIZATION SUBSCRIPTION statement [MobiLink] enhancement** You can now specify a script version and subscription name. See “CREATE SYNCHRONIZATION
  SUBSCRIPTION statement [MobiLink]” [SQL Anywhere Server - SQL Reference].

- **ALTER SYNCHRONIZATION SUBSCRIPTION statement [MobiLink] enhancement** You can use the new SET SCRIPT VERSION clause to specify the script version to use during
  synchronization. You can also use the RENAME clause to rename a subscription. See “ALTER
  SYNCHRONIZATION SUBSCRIPTION statement [MobiLink]” [SQL Anywhere Server - SQL
  Reference].

### Consolidated databases

Following is a list of enhancements to consolidated database support for MobiLink introduced in SQL
Anywhere version 12.0.0.

- **Newly supported consolidated databases** The following new consolidated databases are
  supported by MobiLink in version 12.0.0:
  - IBM DB2 LUW 9.7
  - SQL Anywhere 12

- **Synchronization of spatial data** Synchronization of spatial data types is now supported for the
  following consolidated databases: SQL Anywhere, Oracle, Microsoft SQL Server, IBM DB2 and
  MySQL. See “Synchronization of spatial data” [MobiLink - Server Administration].

- **Support for longer CHAR columns** The MobiLink server now supports synchronizing CHAR,
  VARCHAR, NCHAR and NVARCHAR remote database columns with byte lengths greater than
  32767 bytes.

- **Support for synchronizing columns with the data type of TIMESTAMP WITH TIME
  ZONE** The MobiLink server now supports synchronizing remote database columns with the data
  type of TIMESTAMP WITH TIME ZONE.

### iAnywhere Solutions 12 - Oracle ODBC driver

Following is a list of enhancements to the iAnywhere Solutions 12 - Oracle ODBC driver introduced in
SQL Anywhere version 12.0.0.

- **VARRAY support in packaged procedures** The iAnywhere Solutions 12 - Oracle ODBC driver
  now supports the use of VARRAYs in packaged procedures.
MobiLink server

Following is a list of enhancements to the MobiLink server introduced in SQL Anywhere version 12.0.0.

New mlsrv12 features

- **Server name change to mlsrv12**  
  The MobiLink server has changed from mlsrv11 to mlsrv12.

- **New -cmax option for mlsrv12**  
  This new option is part of the dynamic cache sizing feature. It sets the maximum size for the server memory cache. See “-cmax mlsrv16 option” [MobiLink - Server Administration].

- **New -cinit option for mlsrv12**  
  This new option is part of the dynamic cache sizing feature. It sets the initial size for the server memory cache. See “-cinit mlsrv16 option” [MobiLink - Server Administration].

- **New -cmin option for mlsrv12**  
  This new option is part of the dynamic cache sizing feature. It sets minimum size for the server memory cache. See “-cmin mlsrv16 option” [MobiLink - Server Administration].

- **New protocol option for -x option for mlsrv12**  
  The -x option for mlsrv12 now supports a new oe protocol option that allows you to use the new integrated outbound enabler when using the Relay Server. See “-x mlsrv16 option” [MobiLink - Server Administration].

- **New default for -sl java option for mlsrv12**  
  By default, the MobiLink server now tries to load a server Java VM if one is present instead of a client VM. To override this new default, you can explicitly request a client VM by adding -client to your -sl java options. See “-sl java mlsrv16 option” [MobiLink - Server Administration].

- **New -ca option for mlsrv12**  
  Provide the host name that runs the MobiLink arbiter when using multiple servers with server-initiated synchronization without lightweight polling. See “-ca mlsrv16 option” [MobiLink - Server Administration].

- **New -ftru option for mlsrv12**  
  This option has been added to enable file upload support in the MobiLink server. This option allows you to set the root directory for files to be uploaded with the mlfiletransfer utility. See “-ftru mlsrv16 option” [MobiLink - Server Administration].

- **New metrics for -ppv option for mlsrv12**  
  A number of new metrics have been added to the -ppv option for mlsrv12. See “-ppv mlsrv16 option” [MobiLink - Server Administration].

- **New -rrp option for mlsrv12**  
  The new -rrp option for mlsrv12 was added in conjunction with the new MobiLink Replay utility (mlreplay). The option causes the MobiLink server to run the mlreplay utility and replay all recorded sessions in the given directory when the server starts. See “-rrp mlsrv16 option” [MobiLink - Server Administration].

- **New -rp option for mlsrv12**  
  The new -rp option for mlsrv12 was added in conjunction with the new MobiLink Replay utility (mlreplay). The option is used to specify the directory to which synchronizations are recorded for playback with the mlreplay utility. See “-rp mlsrv16 option” [MobiLink - Server Administration].
New -vR option for mlsrv12  Use the new -vR option for mlsrv12 to show the remote ID in each log message for synchronization. See “-v mlsrv16 option” [MobiLink - Server Administration].

New -vU option for mlsrv12  Use the new -vU option for mlsrv12 to show the MobiLink user name in each log message for synchronization. See “-v mlsrv16 option” [MobiLink - Server Administration].

New -vk option for mlsrv12  This new option is part of the dynamic cache sizing feature. This option prints a line to the log whenever the cache grows or shrinks. See “-v mlsrv16 option” [MobiLink - Server Administration].

Improvements to low memory performance in MobiLink server  There is improved behavior and robustness in low-memory conditions.

Two new conflict detection approaches  The upload_fetch, upload_fetch_column_conflict, upload_new_row_insert, and upload_old_row_insert scripts can be used to detect conflicts. See “Conflict detection” [MobiLink - Server Administration].

New ml_add_missing_dnld_scripts system procedure  The new system procedure can be used to add missing download_cursor and download_delete_cursor scripts. See “ml_add_missing_dnld_scripts system procedure” [MobiLink - Server Administration].

New MobiLink server API features

New TimestampWithTimeZone class added (Java)  This class provides methods that allow you to retrieve and specify the time zone offset of a Timestamp value. See “TimestampWithTimeZone class [MobiLink server Java]” [MobiLink - Server Administration].

New DateTimeWithTimeZone class added (.NET)  This class provides methods that allow you to retrieve and specify the time zone offset of a DateTime value. See “DateTimeWithTimeZone class [MobiLink server .NET]” [MobiLink - Server Administration].

New SpatialUtilities classes added (.NET and Java)  This class provides methods that assist you when working with spatial data. See “SpatialUtilities class [MobiLink server .NET]” [MobiLink - Server Administration] (.NET) and “SpatialUtilities class [MobiLink server Java]” [MobiLink - Server Administration]. (Java)

New MobiLink scripting features

New system parameters available for upload_fetch and upload_fetch_column_conflict scripts  The MobiLink server system parameters remote_id and username are now available for the upload_fetch and upload_fetch_column_conflict scripts.

New parameters are available for authenticate_file_transfer scripts  New parameters are available for authenticate_file_transfer scripts. See “authenticate_file_transfer connection event” [MobiLink - Server Administration].

New authenticate_file_upload connection event  A new connection event is available to support file uploads. See “authenticate_file_upload connection event” [MobiLink - Server Administration].
- **New generate_next_last_download_timestamp script** Use this script instead of modify_next_last_download_timestamp if you want to suppress MobiLink's default algorithm for determining the next last download time. See “generate_next_last_download_timestamp event” [MobiLink - Server Administration].

**Server utilities**

Following is a list of enhancements to the MobiLink server utilities introduced in SQL Anywhere version 12.0.0.

- **New mlarbiter utility** The new mlarbiter utility is used to start the MobiLink arbiter server. The arbiter is required when using server-initiated synchronization with a MobiLink server farm. See “MobiLink Arbiter Server utility for Windows (mlarbiter)” [MobiLink - Server Administration].

**MobiLink Monitor**

Following is a list of enhancements to the MobiLink Monitor introduced in SQL Anywhere version 12.0.0.

- **New entries in the graph pane** The following entries have been added to the graph pane in the MobiLink Monitor: OE work queue, Notifier work queue, and Dynamic Cache work queue. See “How the Utilization Graph works” [MobiLink - Server Administration].

- **New Trusted Certificate File option** A new Trusted Certificate File option and Browse button has been added to the Connect To MobiLink Server window.

**MobiLink clients**

Following is a list of enhancements to MobiLink clients introduced in SQL Anywhere version 12.0.0.

- **Dbmlsync has enhanced support for background synchronization** The database engine can now drop the dbmlsync connection to the remote database (and rollback uncommitted operations dbmlsync has) if another connection is waiting for access to any database resource that dbmlsync has locked. This allows the other connection to go forward without waiting for the synchronization to complete.

  Use these options for background synchronization:

  - 
    - **-bk dbmlsync option** [MobiLink - Client Administration]
    - **-bkr dbmlsync option** [MobiLink - Client Administration]
    - Background and BackgroundRetry options for “MobiLink synchronization profiles” [MobiLink - Client Administration]

- **dbmlsync now supports named subscriptions** The new -s option for dbmlsync allows you to specify the names of subscriptions to be synchronized. See “-s dbmlsync option” [MobiLink - Client Administration].
HTTP response buffering  Use the new network protocol option http_buffer_responses to completely stream HTTP packets from MobiLink into an intermediary buffer before being processed instead of processing the bytes as they are read off the wire. See “http_buffer_responses” [MobiLink - Client Administration].

Client-side certificates can now be used to authenticate MobiLink clients to third party servers and proxies  The identity and identity_password synchronization parameters have been added to provide support for this feature.

Synchronization no longer required to implement schema changes on remote databases (for SQL Anywhere clients only)  You can simplify schema changes on the remote database by using the START and STOP SYNCHRONIZATION SCHEMA CHANGE statements to delimit your schema change. See:

- “START SYNCHRONIZATION SCHEMA CHANGE statement [MobiLink]” [SQL Anywhere Server - SQL Reference]
- “STOP SYNCHRONIZATION SCHEMA CHANGE statement [MobiLink]” [SQL Anywhere Server - SQL Reference]

Use the CREATE or ALTER SYNCHRONIZATION SUBSCRIPTION statement to specify a script version for each synchronization subscription. See:

- “CREATE SYNCHRONIZATION SUBSCRIPTION statement [MobiLink]” [SQL Anywhere Server - SQL Reference]
- “ALTER SYNCHRONIZATION SUBSCRIPTION statement [MobiLink]” [SQL Anywhere Server - SQL Reference]

New mlfileupload method for UltraLite  The MLFileUpload method has been added for UltraLite clients.

Changes to the dbmlsync options window  The following changes have been made to the dbmlsync options window:

- Retry on remote progress, Remote is behind, Remote is ahead, Drop conflicting connections, Site Script, and Cmdline help have been removed.
- The Publication option has been replaced with a Subscription option.

New dbmlsync options

- New -s option for dbmlsync  Use the new -s option for dbmlsync to specify the names of the subscriptions to be synchronized. See “-s dbmlsync option” [MobiLink - Client Administration].

- New -bk option for dbmlsync  This option enables background synchronization. See “-bk dbmlsync option” [MobiLink - Client Administration].

- New -bkr option for dbmlsync  This option controls the behavior of dbmlsync after a background synchronization is interrupted. See “-bkr dbmlsync option” [MobiLink - Client Administration].

- New -ci option for dbmlsync  This option is part of the new dynamic cache sizing feature. Use this option to set the initial size of the cache used by dbmlsync for synchronization data. See “-ci dbmlsync option” [MobiLink - Client Administration].
● **New -cl option for dbmlsync**  This option is part of the new dynamic cache sizing feature. Use this option to set the minimum size to which the dbmlsync cache file will be reduced. See “-cl dbmlsync option” [MobiLink - Client Administration].

● **New -cm option for dbmlsync**  This option is part of the new dynamic cache sizing feature. Use this option to set the maximum size limit for the dbmlsync cache file. See “-cm dbmlsync option” [MobiLink - Client Administration].

● **New BufferDownload extended option for dbmlsync**  This option is part of the new dynamic cache sizing feature. The option specifies whether the entire download from the MobiLink server should be read into the cache before applying it to the remote database. See “BufferDownload (bd) extended option” [MobiLink - Client Administration].

New Dbmlsync C++ API objects

● **CancelSync method**  This method allows MobiLink clients to cancel a synchronization. See “DbmlsyncClient.CancelSync method [Dbmlsync C++]” [MobiLink - Client Administration].

● **DBSC_CancelRet enumeration**  This enumeration indicates the result of a synchronization cancellation attempt. See “DBSC_CancelRet enumeration [Dbmlsync C++]” [MobiLink - Client Administration].

● **DBSC_ERR_ACTIVE_SYNC_NOT_CANCELED member**  This member indicates that the server could not cancel the synchronization request because the synchronization was active. See “DBSC_ErrorType enumeration [Dbmlsync C++]” [MobiLink - Client Administration].

● **DBSC_ERR_DEAD_SERVER member**  This member indicates that the dbmlsync server encountered an error while starting up and is shutting down. See “DBSC_ErrorType enumeration [Dbmlsync C++]” [MobiLink - Client Administration].

● **Enable status property**  Three new events (DBSC_EVENTTYPE_ML_CONNECT, DBSC_EVENTTYPE_UPLOAD_COMMITED, and DBSC_EVENTTYPE_DOWNLOAD_COMMITED) are sent to the MobiLink client when this property is enabled. See “DBSC_EventType enumeration [Dbmlsync C++]” [MobiLink - Client Administration], “DbmlsyncClient.SetProperty method [Dbmlsync C++]” [MobiLink - Client Administration] and “DbmlsyncClient.GetProperty method [Dbmlsync C++]” [MobiLink - Client Administration].

New Dbmlsync .NET API objects

● **CancelSync method**  This method allows MobiLink clients to cancel a synchronization. See “DbmlsyncClient.CancelSync method [Dbmlsync .NET]” [MobiLink - Client Administration].

● **DBSC_CancelRet enumeration**  This enumeration indicates the result of a synchronization cancellation attempt. See “DBSC_CancelRet enumeration [Dbmlsync .NET]” [MobiLink - Client Administration].

● **DBSC_ERR_ACTIVE_SYNC_NOT_CANCELED member**  This member indicates if the server could not cancel the synchronization request if the synchronization was active. See “DBSC_ErrorType enumeration [Dbmlsync .NET]” [MobiLink - Client Administration].
● **DBSC_ERR_DEAD_SERVER** member This member indicates that the dbmlsync server encountered an error while starting up and is shutting down. See “DBSC_ErrorType enumeration [Dbmlsync .NET]” [MobiLink - Client Administration].

● **"enable status" property** Three new events (DBSC_EVENTTYPE_ML_CONNECT, DBSC_EVENTTYPE_UPLOAD_COMMITTED, and DBSC_EVENTTYPE_DOWNLOAD_COMMITTED) are sent to the MobiLink client when this property is enabled. See “DBSC_EventType enumeration [Dbmlsync .NET]” [MobiLink - Client Administration], “DbmlsyncClient.SetProperty method [Dbmlsync .NET]” [MobiLink - Client Administration] and “DbmlsyncClient.GetProperty method [Dbmlsync .NET]” [MobiLink - Client Administration].

New client utilities features

● **Upload files using mlfiletransfer** You can now upload files using the mlfiletransfer utility. See “MobiLink File Transfer utility (mlfiletransfer)” [MobiLink - Client Administration].

● **New -i option for mlfiletransfer utility** The -i option was added to mlfiletransfer to disable resume functionality. See “MobiLink File Transfer utility (mlfiletransfer)” [MobiLink - Client Administration].

**Relay Server**

The following Relay Server features have been added in this release:

● **Support for SQL Anywhere Monitor** Relay Server resources can now be monitored using the SQL Anywhere Monitor. See “SQL Anywhere Monitor” [SQL Anywhere Server - Database Administration].

● **Support for central administration** The Relay Server supports central administration via Sybase Central.

● **RSOE supported on Mac OS** The Relay Server Outbound Enabler is now supported on Mac OS. For more detailed information about supported platforms, see http://www.sybase.com/detail?id=1061806.

● **Windows IIS 7 and 7.5 now supported** The Relay Server is now supported for IIS 7 on Windows Server 2008 and IIS 7.5 on Windows Server 2008 R2. For more detailed information about supported platforms, see http://www.sybase.com/detail?id=1061806.

● **New active_cookie option** The active_cookie option has been added to the backend farm section of the Relay Server configuration. See “Backend farm section properties” [Relay Server].

● **New active_header option** The active_header option has been added to the backend farm section of the Relay Server configuration. See “Backend farm section properties” [Relay Server].

● **Improvements to Relay Server troubleshooting** To improve troubleshooting, the Relay Server now has standardized Relay Server error codes with localized error messages and selective error messages with system error codes and embedded system error messages. See “Relay Server error messages” [Error Messages].
- **Support for SQL Anywhere database as a back-end HTTP server**  The Relay Server now supports SQL Anywhere failover and read-only scale out. See “SQL Anywhere web services high availability and scale-out solutions” [SQL Anywhere Server - Database Administration].

- **Outbound Enabler dynamic response buffer sizing**  Dynamic response buffer sizing has significantly reduced the Outbound Enabler memory overhead.

- **Outbound Enabler user interface improvements**  An instance identifier in window title, systray tip text and systray menu have all been added to the Outbound Enabler user interface, and the Outbound Enabler status has been added to the systray tip text.

- **More efficient use of shared memory**  The Relay Server uses shared memory (set by the shared_mem configuration option) more efficiently. In deployments with relatively slow reading clients exercising HTTP requests with large responses, the relay server is able to run with significantly less shared memory.

- **Relay Server Outbound Enabler now supports periodic back-end server status requests using HTTP**  The RSOE has been enhanced to support periodic back-end server status requests using HTTP. A back-end server status request is an alternative to the liveness ping and can be used to determine if the back-end server is able to accept client requests or not. The new status_url parameter, which is specified as part of the rsoe -cs option, is used to enable periodic back-end server status requests.

---

**MobiLink behavior changes**

Following is a list of behavior changes to MobiLink introduced in version 12.0.0. For information about supported platforms and versions, see [http://www.sybase.com/detail?id=1061806](http://www.sybase.com/detail?id=1061806).

**MobiLink server changes**

- **MobiLink server no longer requires log4j.jar**  The log4j.jar file is no longer required by the MobiLink server and is no longer deployed with the MobiLink server. If you require log4j.jar you must install your own version of the jar and put it in the classpath.

- **New behavior for -cn option**  The -cn option for mlsrv12 sets the maximum number of database connections used for database worker threads. In versions prior to SQL Anywhere 12, the -cn option for mlsrv12 set the maximum number of database connections. See “-cn mlsrv16 option” [MobiLink - Server Administration].

- **New behavior for -sl dnet option**  Previously, the MobiLink server loaded the workstation CLR by default when using .NET scripts. It now loads the server CLR instead. You can restore the old behavior by adding -clrFlavor=wks to the -sl dnet option for mlsrv12. See “-sl dnet mlsrv16 option” [MobiLink - Server Administration].

- **MobiLink server now uses the GV_$TRANSACTION Oracle system view instead of V_$TRANSACTION**  The Oracle account used by the MobiLink server must now have permission for the GV_$TRANSACTION Oracle system view instead of the V_$TRANSACTION system view. See “Oracle consolidated database” [MobiLink - Server Administration].
• **Upgrade scripts for version 6.0.x removed** The MobiLink upgrade scripts for 6.0.x have been removed. If you require this upgrade, contact Technical Support (http://www.sybase.com/support).

• **ml_add_column system procedure no longer required** For version 12 or later clients, the ml_add_column system procedure is no longer required when you want to use column names in named parameters. By default, you can now reference the column names directly without any extra setup. See “ml_add_column system procedure (deprecated)” [MobiLink - Server Administration].

• **New BIGINT data type support for Java and .NET server APIs** The BIGINT SQL data type now maps to the LONG Java and .NET data types. See “SQL-Java data types” [MobiLink - Server Administration] and “SQL-.NET data types” [MobiLink - Server Administration].

• **Data scripts are now required** To reduce the chance of losing data from accidentally not creating data scripts, the MobiLink server now requires either an ignored script or a valid script for the following events. See “Ignored scripts” [MobiLink - Server Administration].
  ○ **upload_insert** if any inserted rows are uploaded from remotes and no handle_UploadData connection script is defined.
  ○ **upload_update** if any updated rows are uploaded from remotes and no handle_UploadData connection script is defined.
  ○ **upload_delete** if any deleted rows are uploaded from remotes and no handle_UploadData connection script is defined.
  ○ **download_cursor and download_delete_cursor** if no handle_DownloadData connection script is defined and the synchronization is not upload-only.

As a convenience when upgrading to version 12, you can use the ml_add_missing_dnld_scripts stored procedure to add ignored scripts to avoid errors from missing download scripts. See “ml_add_missing_dnld_scripts system procedure” [MobiLink - Server Administration].

### MobiLink client changes

Following is a list of behavior changes to MobiLink clients introduced in version 12.0.0.

• **Version 12 MobiLink synchronization clients send column names by default** In version 12, MobiLink synchronization clients all send column names to the MobiLink server by default. This means that in most cases, you are no longer required to use the ml_add_column system stored procedure to define column names and ordering for use with named parameters in MobiLink scripts. See “ml_add_column system procedure (deprecated)” [MobiLink - Server Administration].

• **Added support for enhanced TLS session renegotiation** MobiLink clients now support a new TLS extension that allows vulnerable third-party servers to be secured.

• **MLFileTransfer method renamed** The MLFileTransfer method for UltraLite clients has been split into the MLFileUpload and MLFileDownload methods.

• **-df has been renamed to -lf** The -df option for the mlfiletransfer utility has been renamed to -lf and now refers to a local file instead of a destination file.
- **-dp has been renamed to -lp** The -dp option for the mlfiletransfer utility has been renamed to -lp and now refers to a local path instead of a destination path.

- **UPLD_ERR_USERID_ALREADY_IN_USE has been changed** Dbmlsync no longer returns UPLD_ERR_USERID_ALREADY_IN_USE as a failure cause for the sp_hook_dbmlsync_upload_end event hook. In its place, the value UPLD_ERR_REMOTE_ID_ALREADY_IN_USE is returned.

- **Palm no longer supported for UltraLite clients** Palm OS is no longer supported. If you want to use Palm, you should continue to use SQL Anywhere 11.

### MobiLink plug-in for Sybase Central changes

Following is a list of behavior changes for the MobiLink plug-in for Sybase Central introduced in version 12.0.0.

- **Changes to Mapping editors for Synchronization models**
  - The Table and Column Mapping editors for synchronization models now show consolidated databases on the left and remote databases on the right. See “Table and column mappings” [MobiLink - Getting Started].
  - Only the synchronized consolidated tables are listed in the table mapping editor. Changing the table mapping direction to **Not Synchronized** is the same as deleting the table mapping, removing that row from the editor when the model is saved, without changing database schema.
  - To add a table mapping for an unsynchronized consolidated table, and optionally add the table to the remote schema, use the **New Table Mappings** command. You can also use the **Update Schema** command to change the model's consolidated or remote schema.
  - Popup menus are now used instead of dropdown lists for choosing the right side of table and column mappings, with an optional window (accessed with the ellipsis button) if all the choices do not fit in the menu. For table mappings, only unsynchronized remote tables are listed. For column mappings, unsynchronized columns in the remote table are listed along with the options for value mappings and for not synchronizing the column.
  - If you change a synchronization option for a table mapping, the lower pane automatically switches to the tab for the corresponding sub-options.

- **VARBIT and LONG VARBIT columns** VARCHAR and LONG VARCHAR columns are used in place of VARBIT and LONG VARBIT columns, respectively, when deploying a synchronization model with a remote schema to a new UltraLite remote database.

- **GO now used as the statement delimiter** For SQL Anywhere and UltraLite databases, synchronization models now use GO for the statement delimiter instead of a semicolon, allowing the SQL generated with synchronization models to be used in central administration of remote tasks.
MobiLink deprecated and discontinued features

Note
As with all forward-looking statements, the lists of deprecated features are not guaranteed to be complete and are subject to change.

- **IBM DB2 Mainframe not supported in Version 12.0.0 or later**  
  IBM DB2 mainframe is not supported as a consolidated database in version 12.0.0. However, MobiLink still supports DB2 LUW (Linux, Unix, and Windows) as a consolidated database.

- **Adaptive Server Enterprise 12.5.x no longer supported**  
  Adaptive Server Enterprise 12.5.x is no longer supported by MobiLink in version 12.0.0.

- **IBM DB2 LUW 8.2 no longer supported**  
  IBM DB2 LUW 8.2 is no longer supported by MobiLink in version 12.0.0.

- **-xo option for mlsrv12 has been removed**  
  Clients earlier than version 10 are no longer supported.

- **-f option for mlsrv12 has been removed**  
  Use the -zf mlsrv12 option to specify that the MobiLink server should check for script changes at the beginning of each synchronization. See “-zf mlsrv16 option” [MobiLink - Server Administration].

- **-nba option for mlsrv12 has been removed**  
  Blocking download acknowledgement is no longer supported. If the remote database requests a download acknowledgement, the MobiLink server will always use a non-blocking acknowledgement.

- **-fr option for mlsrv12 has been removed**  
  The -fr option for mlsrv12 is no longer supported. If you want to ignore a script (which might cause data to be lost) then define the script as --{ml_ignore}.

- **Java and .NET data scripts returning SQL is deprecated**  
  The ability for Java and .NET scripting logic to return strings that are interpreted by MobiLink server as SQL scripts is deprecated in data scripts. If your scripts need to cause changes in the consolidated database, they should do so directly from Java or .NET.

  See:
  - “Java and .NET data scripts returning SQL (removed)” [MobiLink - Server Administration]
  - “Direct row handling” [MobiLink - Server Administration]
  - “Data scripts” [MobiLink - Server Administration]

- **Download error hooks removed**  
  The following dbmlsync hooks have been removed in version 12: sp_hook_dbmlsync_download_com_error, sp_hook_dbmlsync_download_fatal_sql_error, and sp_hook_dbmlsync_download_sql_error.

- **-f option for MobiLink File Transfer utility has been removed**  
  The -f option for the mlfiletransfer utility is no longer supported.

- **-r option for MobiLink File Transfer utility has been removed**  
  The -r option for the mlfiletransfer utility is no longer supported.
Memory (mem) and DownloadBufferSize (dbs) extended options for dbmlsync have been removed
The Memory (mem) and DownloadBufferSize (dbs) extended options for dbmlsync is are no longer supported. Use the CacheMin, CacheInit and CacheMax options instead.

dbmlsync support for SQL passthrough has been removed
The SQL passthrough feature is no longer supported for MobiLink clients. It has been replaced by the central administration of remote databases feature. See “Central administration of remote databases” [MobiLink - Server Administration].

-ss option for mlsrv12 no longer required
Prior to version 12, the -ss option for mlsrv12 was used enable the MobiLink server to run in a server farm environment. New remote ID locking logic that prevents redundant synchronizations, the -ss option is no longer necessary for MobiLink servers running in a server farm, and the option has been removed. An arbiter is required when using server-initiated synchronization with a MobiLink server farm.

Note
Running the MobiLink server in a server farm is a feature of the MobiLink high availability option, which requires a separate license. See “Separately licensed components” [SQL Anywhere 16 - Introduction].

The MobiLink Redirector has been removed
The Redirector is no longer available. Use the Relay Server instead. See “Introduction to the Relay Server” [Relay Server].

Recommendation for using script versions
It is highly recommended that you no longer use the ScriptVersion extended option. Instead, use the SCRIPT VERSION clause on the CREATE SYNCHRONIZATION SUBSCRIPTION and ALTER SYNCHRONIZATION SUBSCRIPTION statements. See “CREATE SYNCHRONIZATION SUBSCRIPTION statement [MobiLink]” [SQL Anywhere Server - SQL Reference] and “ALTER SYNCHRONIZATION SUBSCRIPTION statement [MobiLink]” [SQL Anywhere Server - SQL Reference].

The -n option for dbmlsync has been deprecated
This option has been deprecated. It is recommended that you use the -s dbmlsync option instead. See “-s dbmlsync option” [MobiLink - Client Administration].

The -u option for dbmlsync has been deprecated
This option has been deprecated. It is recommended that you use the -s dbmlsync option instead. See “-s dbmlsync option” [MobiLink - Client Administration].

The Publication synchronization profile option has been deprecated
This option has been deprecated. It is recommended that you use the -s dbmlsync option instead. See “-s dbmlsync option” [MobiLink - Client Administration].

The MLUser synchronization profile option has been deprecated
This option has been deprecated. It is recommended that you use the -s dbmlsync option instead. See “-s dbmlsync option” [MobiLink - Client Administration].

The dbmlsync Integration component has been removed
The Dbmlsync integration component has been removed. In its place, use the dbmlsync programming interface. See “Dbmlsync API” [MobiLink - Client Administration].
Forced-conflict mode has been deprecated  The MobiLink server uses forced conflict resolution when the upload_insert, upload_update, and upload_delete script are all undefined. This feature has been deprecated.

Detecting conflicts with upload_update has been deprecated  You should either detect and resolve uploaded update conflicts in your upload_update script or use either an upload_fetch or upload_fetch_column_conflict script to detect conflicts. Relying on the MobiLink server to count affected rows by your upload_update script to detect a conflict, and then invoke your conflict resolution scripts, has been deprecated. See “Conflict detection with upload_update scripts” [MobiLink - Server Administration].

The use of question marks in SQL scripts  The use of plain question marks in MobiLink SQL scripts has been deprecated. Use named parameters instead. See “Named script parameters” [MobiLink - Server Administration].

SQL Remote new features

Following is a list of additions to SQL Remote introduced in version 12.0.0. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

- SQL Remote now supports replication of spatial values  SQL Remote now supports replication of spatial data types. In addition, SQL Remote supports the replication of the new TIMESTAMP WITH TIMEZONE data type. See “TIMESTAMP WITH TIME ZONE data type” [SQL Anywhere Server - SQL Reference].

- -g option added to Extraction utility (dbxtract)  By default, materialized views defined as MANUAL REFRESH are not initialized as part of a reload. You can now use the -g option with dbxtract to initialize these materialized views as part of the reload process. See “Extraction utility (dbxtract)” [SQL Remote].

- SQL Remote now prints an error message when no publisher is defined  When SQL Remote connects to a database that has remote or consolidated users defined, but no publisher defined, then SQL Remote returns an error indicating that no publisher is defined in the database.

UltraLite new features

Following is a list of additions to UltraLite introduced in version 12.0.0. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

General features

- Support for spatial data  The following features have been added in support of the new spatial data capabilities in UltraLite:
ST_Geometry data type  The ST_Geometry data type supports functions that can be applied to spatial values. See “ST_GEOMETRY data type” [UltraLite - Database Management and Reference].

Spatial data functions  UltraLite provides several functions to support the processing of spatial data. See “ST_GEOMETRY data type” [UltraLite - Database Management and Reference].

New encryption examples  New code samples have been added to illustrate UltraLiteJ encryption on BlackBerry smartphones. For more information, see the %SQLANYSAMP12%UltraLiteJ directory.

Platforms and devices

iPhone and Mac OS X support
UltraLite applications can now be developed on Mac OS X to target Mac OS X and iPhone using the UltraLite C/C++ API. For more information, see “UltraLite C++ application development” [UltraLite - C and C++ Programming]. There is also a detailed tutorial to assist you in developing an iPhone application.

64-bit Linux installs
If you install SQL Anywhere on a 64-bit Linux computer, you must also install the 32-bit subsystem separately. This is because some 64-bit Linux operating systems do not include pre-installed 32-bit compatibility libraries. To use 32-bit client software, you may need to install 32-bit compatibility libraries for your Linux distribution. For example, on Ubuntu, you may need to run the following command:

```
sudo apt-get install ia32-libs
```

Security

UltraLite supports FIPS 140-2 certified encryption  FIPS 140-2 certified encryption is now supported for UltraLite on 64-bit Windows.

UltraLite database encryption now uses 256-bit AES  UltraLite database encryption is now performed using 256-bit AES instead of 128-bit AES.

Salt is now used when hashing passwords  A random 4-byte salt value is now generated every time a new user is created or an existing user changes their password. See “UltraLite PWD connection parameter” [UltraLite - Database Management and Reference].

Utilities

UltraLite utilities improvements  The ulinit utility has been enhanced to allow you to create an UltraLite database based on information in a SQL Anywhere database even when the schema being extracted from the SQL Anywhere database contains elements that UltraLite does not support (such as column datatypes or default values for instance). This is now the default behavior (which can be
turned off using the -f option). See “UltraLite Initialize Database utility (ulinit)” [UltraLite - Database Management and Reference]. The ulcreate utility is no longer supported.

Changes have also been made to the ulinit, ulerase, ulinfo, ulload, and ulsync utilities.

See:

- “UltraLite Initialize Database utility (ulinit)” [UltraLite - Database Management and Reference]
- “UltraLite Erase utility (ulerase)” [UltraLite - Database Management and Reference]
- “UltraLite Information utility (ulinfo)” [UltraLite - Database Management and Reference]
- “UltraLite Load XML to Database utility (ulload)” [UltraLite - Database Management and Reference]
- “UltraLite Synchronization utility (ulsync)” [UltraLite - Database Management and Reference]

## SQL

- **CREATE / DROP / ALTER USER added**  
  See:
  - “CREATE USER statement [UltraLite]” [UltraLite - Database Management and Reference]
  - “DROP USER statement [UltraLite]” [UltraLite - Database Management and Reference]
  - “ALTER USER statement [UltraLite]” [UltraLite - Database Management and Reference]

- **New table constraint for CREATE TABLE and ALTER TABLE**  
  An additional table constraint (SYNCHRONIZE ON | OFF | ALL) can be specified in a CREATE TABLE or an ALTER TABLE statement. See:
  - “CREATE TABLE statement [UltraLite]” [UltraLite - Database Management and Reference]
  - “ALTER TABLE statement [UltraLite]” [UltraLite - Database Management and Reference]

- **IF EXISTS clause added to DROP statements**  
  A new IF EXISTS clause can now optionally be specified in DROP INDEX, DROP PUBLICATION, DROP SYNCHRONIZATION PROFILE, and DROP TABLE statements. See:
  - “DROP INDEX statement [UltraLite]” [UltraLite - Database Management and Reference]
  - “DROP PUBLICATION statement [UltraLite]” [UltraLite - Database Management and Reference]
  - “DROP SYNCHRONIZATION PROFILE statement [UltraLite]” [UltraLite - Database Management and Reference]
  - “DROP TABLE statement [UltraLite]” [UltraLite - Database Management and Reference]
UltraLite new features

- **IF NOT EXISTS clause added to CREATE statements**  A new IF NOT EXISTS clause can now optionally be specified in CREATE TABLE, CREATE INDEX, CREATE PUBLICATION, and CREATE SYNCHRONIZATION PROFILE statements. See:
  - “CREATE TABLE statement [UltraLite]” [UltraLite - Database Management and Reference]
  - “CREATE INDEX statement [UltraLite]” [UltraLite - Database Management and Reference]
  - “CREATE PUBLICATION statement [UltraLite]” [UltraLite - Database Management and Reference]
  - “CREATE SYNCHRONIZATION PROFILE statement [UltraLite]” [UltraLite - Database Management and Reference]

- **CREATE SYNCHRONIZATION PROFILE now also supports OR REPLACE clause**  The CREATE SYNCHRONIZATION PROFILE statement now also support the option to replace a table. See “CREATE SYNCHRONIZATION PROFILE statement [UltraLite]” [UltraLite - Database Management and Reference].

- **COUNT_UPLOAD_ROWS function added**  COUNT_UPLOAD_ROWS allows you to query the number of rows that will be uploaded in the next synchronization. See “COUNT_UPLOAD_ROWS function [Aggregate]” [UltraLite - Database Management and Reference].

### Data types

- **ST_Geometry**  The ST_Geometry data type supports functions that can be applied to spatial values. See “ST_GEOMETRY data type” [UltraLite - Database Management and Reference].

- **TIMESTAMP WITH TIMEZONE data type**  The TIMESTAMP WITH TIMEZONE data type allows date and time values to be stored together with time zone offsets. A time zone offset is the number of minutes before or after UTC. See “UltraLite Initialize Database utility (ulinit)” [UltraLite - Database Management and Reference] and “UltraLite timestamp_with_time_zone_format creation parameter” [UltraLite - Database Management and Reference].

### Programming interfaces

**UltraLite C/C++**

- **New UltraLite C/C++ API added**  A new version of the UltraLite C/C++ API has been added to this release. This API is declared in the ulcpp.h header file. See “UltraLite C/C++ API reference” [UltraLite - C and C++ Programming].
The following objects have been renamed for this version of the UltraLite C/C++ API:

- `ul_synch_info` has been renamed to `ul_sync_info`.
- `ul_synch_result` has been renamed to `ul_sync_result`.
- `ul_synch_status` has been renamed to `ul_sync_status`.
- `ul_synch_stats` has been renamed to `ul_sync_stats`.
- `ul_synch_observer_fn` has been renamed to `ul_sync_observer_fn`.
- `ul_synch_state` has been renamed to `ul_sync_state`.
- `ULInitSynchInfo` has been renamed to `ULInitSyncInfo`.
- `ULSetSynchInfo` has been renamed to `ULSetSyncInfo`.
- `ULGetSynchResult` has been renamed to `ULGetSyncResult`.
- All `UL_SYNCH_STATE` objects have been renamed to `UL_SYNC_STATE` objects.
- `UL_SYNCH_STATUS_FLAG_IS_BLOCKING` has been renamed to `UL_SYNC_STATUS_FLAG_IS_BLOCKING`.
- `ULRegisterErrorCallback` has been renamed to `ULSetErrorCallback`.
- `ULRegisterSynchronizationCallback` has been renamed to `ULSetSynchronizationCallback`.

The following methods have been removed from the C++ API: `GetDatabaseID`, `SetDatabaseID`, `IsCaseSensitive`, and `GetCollationName`. The functionality is now handled by `GetDatabaseProperty` and `SetDatabaseOption`. See “ULConnection.GetDatabaseProperty method [UltraLite C++]” [UltraLite - C and C++ Programming] and “ULConnection.SetDatabaseOption method [UltraLite C++]” [UltraLite - C and C++ Programming].

The `GetDatabasePropertyInt` method has been added. See “ULConnection.GetDatabasePropertyInt method [UltraLite C++]” [UltraLite - C and C++ Programming].

**UltraLite.NET**

- **New ULConnection.ValidateDatabase(ULDBValid) method added** This method has been added as an equivalent to calling `ULConnection.ValidateDatabase(ULDBValid, String)` while passing `null` for the `tableName`. See “ULConnection.ValidateDatabase(ULDBValid) method [UltraLite.NET]” [UltraLite - .NET Programming].

**UltraLite for M-Business Anywhere**

- **API features** The following objects have been added to this release:
  - The `setPublications` method in the `SyncParms` class. This method sets the publication to be synchronized.
  - The `getPublications` method in the `SyncParms` class. This method returns the publication to be synchronized.
  - The `timestampWithTimeZoneFormat` member in the `CreationParms` class. This member sets the format for the timestamp with the time zone retrieved from the database.
  - The `setTimestampWithTimeZoneParameter` method in the `PreparedStatement` class and the `setTimestampWithTimeZone` method in the `ULTable` class. These methods set the value for the specified `SQLType.TIMESTAMP_WITH_TIME_ZONE` type parameter using a Date object.
The getTimestampWithTimeZone methods in the ResultSet and ULTable classes. These methods return the value for the specified column as a Date object.

## UltraLite behavior changes and deprecated features

### Note
As with all forward-looking statements, the lists of deprecated features are not guaranteed to be complete and are subject to change.

Following is a list of deprecated features and behavior changes to UltraLite introduced in version 12.0.0. For information about supported platforms and versions, see [http://www.sybase.com/detail?id=1061806](http://www.sybase.com/detail?id=1061806).

### Deprecated features

- **ulcreate utility no longer supported**  The ulcreate utility is no longer available. All of the functionality is now handled by ulinit. See “UltraLite Initialize Database utility (ulinit)” [UltraLite - Database Management and Reference].

- **SQL passthrough no longer supported**  SQL passthrough, originally released with UltraLite 11, is no longer supported. This functionality is now handled by the “Central administration of remote databases” [MobiLink - Server Administration].

- **Option to ALTER SYNCHRONIZATION PROFILE using the OR REPLACE clause is no longer supported**  The OR REPLACE clause has been removed from the ALTER SYNCHRONIZATION PROFILE statement.

- **UltraLite ODBC API no longer supported**  The UltraLite ODBC API is no longer supported. Use the UltraLite C/C++ API in its place. See “UltraLite - C and C++ Programming”.

### Deprecated platforms

- **UltraLite for M-Business Anywhere no longer supported**  UltraLite support for M-Business Anywhere is deprecated for UltraLite 12.

- **Palm operating system no longer supported**  The Palm operating system is not supported by UltraLite 12.

### Removed, deprecated, and modified APIs

- **Replaced UltraLite C/C++ API**  The UltraLite C/C++ API defined in the uliface.h header file has been replaced by a new version that is defined in the ulcpp.h header file. The previous version of the API is still available. For documentation on the deprecated UltraLite C/C++ API, see [http://dcx.sybase.com/1101en/ulc_en11/c-common-apiref.html](http://dcx.sybase.com/1101en/ulc_en11/c-common-apiref.html).

You can use the old implementation of the UltraLite C/C++ API by adding the %SQLANY11%\$DK \ulcpp11.cpp file to your UltraLite application project, where SQLANY11 is an environment variable that points to your SQL Anywhere installation directory.

- **Modified UltraLite C/C++ API objects**  The following objects have been modified since the last release and apply to the new UltraLite C/C++ API:
○ SQL Passthrough is no longer supported by the API. The following objects have been removed:

- `ul_sql_passthrough_state`
- `ul_sql_passthrough_status`
- `ul_sql_passthrough_observer_fn`

**Modified UltraLite C/C++ common API objects** The following objects have been modified since the last release:

○ The `MLFileTransfer` function has been renamed to `MLFileDownload`. See “MLFileDownload method [UltraLite Embedded SQL]” [UltraLite - C and C++ Programming].

○ The `force_transfer` field of `ml_file_transfer_info` structure has been removed.

○ The `enable_resume` field of `ml_file_transfer_info` now defaults to true instead of false. See “MLFileDownload method [UltraLite Embedded SQL]” [UltraLite - C and C++ Programming].

○ `MLFileDownload` supports the new `remote_key` field of `ml_file_transfer_info` which is passed to MobiLink server scripts to allow greater control of file transfers. See “MLFileDownload method [UltraLite Embedded SQL]” [UltraLite - C and C++ Programming].

**embedded Visual C++ not supported as of UltraLite 11.0** Support for Visual Studio 2003 ended with UltraLite 11.0. Support for embedded Visual C++ was therefore moved into Visual Studio 2005.

**Modified Embedded SQL API objects** The following objects have been modified since the last release:

○ SQL Passthrough is no longer supported by the API. The following objects have been removed:

- `ULGetSQLPassthroughScriptCount`
- `ULExecuteNextSQLPassthroughScript`
- `ULExecuteSQLPassthroughScripts`
- `ULSetSQLPassthroughCallback`

**Modified UltraLite for M-Business Anywhere API objects** The following objects have been modified since the last release:

○ The `GetSQLPassthroughScriptCount`, `ExecuteNextSQLPassthroughScript`, and `ExecuteSQLPassthroughScripts` methods in the `Connection` class have been removed.

○ The syntax of the `CreateDatabase` method in the `DatabaseManager` class has changed.

**Modified UltraLite.NET API objects** The following objects have been modified since the last release:

○ The `DatabaseManager` property under the `ULConnection` class has been removed and is no longer required. `ULDatabaseManager` is no longer a singleton class; the methods are now static. See “ULDatabaseManager class [UltraLite.NET]” [UltraLite - .NET Programming].
○ The DatabaseOnCE property in the ULConnectionParms class has been renamed to DatabaseOnDevice. See “ULConnectionParms.DatabaseOnDevice property [UltraLite.NET]” [UltraLite - .NET Programming].

○ The GetOptimalIndex method in the ULTableSchema class now returns the name of the optimal index. See “ULTableSchema.GetOptimalIndex method [UltraLite.NET]” [UltraLite - .NET Programming].

○ The CountUploadRows(String, UInt32) method in the ULConnection class has been removed. Use CountUploadRows(String, Int64) instead. See “ULConnection.CountUploadRows method [UltraLite.NET]” [UltraLite - .NET Programming].

○ SQL Passthrough is no longer supported by the API. The following objects have been removed:
  ● ULConnection.GetSQLPassthroughScriptCount
  ● ULConnection.ExecuteNextSQLPassthroughScript
  ● ULConnection.ExecuteSQLPassthroughScripts
  ● ULSqlPassthroughProgressListener
  ● ULSqlProgressData
  ● ULSqlProgressState

○ The ULPublicationSchema class and its methods have been removed, along with the GetPublicationSchema method in the ULDatabaseSchema class. The SYNC_ALL_DB and SYNC_ALL_PUBS fields have moved to the ULConnection class. See “ULConnection.SYNC_ALL_DB field [UltraLite.NET]” [UltraLite - .NET Programming] and “ULConnection.SYNC_ALL_PUBS field [UltraLite.NET]” [UltraLite - .NET Programming].

Miscellaneous

● User publication limit increase The maximum number of user publications has increased to 63.

● Default encoding for UltraLite databases is now UTF-8 encoded UltraLite databases are now UTF-8 encoded by default. See “UltraLite utf8_encoding creation parameter” [UltraLite - Database Management and Reference].

UltraLiteJ new features

Following is a list of additions to UltraLiteJ introduced in version 12.0.0. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

● Support for external BLOB files UltraLiteJ, using BlackBerry OS or Java SE, supports the partitioning of database files such that external files may now be used to store large BLOB values, with the files referenced using specific columns in the database. This is implemented as part of the CREATE TABLE SQL function.

A sample usage scenario might include having a BlackBerry application use the smaller but faster persistent store to store an UltraLiteJ database while storing large BLOB values, such as pictures, in larger (but slower) flash drives or SD cards. An added benefit is that applications that capture pictures
and store them in the database do not waste battery power and time copying the pictures into the database.

See “CREATE TABLE statement [UltraLite]” [UltraLite - Database Management and Reference].

- **BlackBerry internal flash and SD cards support** UltraLiteJ can read and write UltraLiteJ databases to either internal flash memory or SD cards.

- **Support for multiple versions of UltraLiteJ** In order to allow multiple versions of UltraLiteJ to co-exist in the BlackBerry and Java ME environment, UltraLiteJ JAR files, COD files, and the Java package name now include the major version number, as follows:
  - The JAR file is called UltraLiteJ12.jar
  - The COD files are called UltraLiteJ12.cod
  - All public classes are contained in package com.ianywhere.ultralitej12
  - The class Unsigned64 has been moved to package com.ianywhere.ultralitej12

- **Encryption and securing UltraLiteJ applications** A new sample demonstrating how to write a very secure UltraLiteJ application for BlackBerry smartphones has been added. See Samples\UltraLiteJ\BlackBerryEncryption\ReadMe.html for details. For a more complete discussion on BlackBerry security, read the UltraLiteJ Security on BlackBerry Devices white paper at http://www.sybase.com/detail_list?id=9814.

- **new ST_Geometry data type** The ST_Geometry data type supports functions that can be applied to spatial values. See “ST_GEOMETRY data type” [UltraLite - Database Management and Reference].

- **TIMESTAMP WITH TIMEZONE data type** The TIMESTAMP WITH TIMEZONE data type allows date and time values to be stored together with time zone offsets. A time zone offset is the number of minutes before or after UTC. See “UltraLite Initialize Database utility (ulinit)” [UltraLite - Database Management and Reference] and “UltraLite timestamp_with_time_zone_format creation parameter” [UltraLite - Database Management and Reference].

- **RAND SQL function support** UltraLiteJ supports the RAND SQL function. See “RAND function [Numeric]” [UltraLite - Database Management and Reference].

- **Support for CREATE SYNCHRONIZATION PROFILE, ALTER SYNCHRONIZATION PROFILE, DROP SYNCHRONIZATION PROFILE, and SYNCHRONIZE** These statements are intended to provide an alternative way to organize sync parameters and to launch synchronizations using SQL. The existing Connection.createSyncParm(), Connection.synchronize(SyncParm) and related APIs continue to work. See:
  - “CREATE SYNCHRONIZATION PROFILE statement [UltraLite]” [UltraLite - Database Management and Reference]
  - “ALTER SYNCHRONIZATION PROFILE statement [UltraLite]” [UltraLite - Database Management and Reference]
  - “DROP SYNCHRONIZATION PROFILE statement [UltraLite]” [UltraLite - Database Management and Reference]
  - “SYNCHRONIZE statement [UltraLite]” [UltraLite - Database Management and Reference]
● **New table constraint for CREATE TABLE and ALTER TABLE**  An additional table constraint (SYNCHRONIZE ON | OFF | ALL) can be specified in a CREATE TABLE or an ALTER TABLE statement. See:

  ○ “CREATE TABLE statement [UltraLite]” [UltraLite - Database Management and Reference]
  ○ “ALTER TABLE statement [UltraLite]” [UltraLite - Database Management and Reference]

● **IF EXISTS clause added**  A new IF EXISTS clause can now optionally be specified in DROP INDEX, DROP PUBLICATION, DROP SYNCHRONIZATION PROFILE, and DROP TABLE statements. See:

  ○ “DROP INDEX statement [UltraLite]” [UltraLite - Database Management and Reference]
  ○ “DROP PUBLICATION statement [UltraLite]” [UltraLite - Database Management and Reference]
  ○ “DROP SYNCHRONIZATION PROFILE statement [UltraLite]” [UltraLite - Database Management and Reference]
  ○ “DROP TABLE statement [UltraLite]” [UltraLite - Database Management and Reference]

● **IF NOT EXISTS clause added**  A new IF NOT EXISTS clause can now optionally be specified in CREATE TABLE, CREATE INDEX, CREATE PUBLICATION, and CREATE SYNCHRONIZATION PROFILE statements. See:

  ○ “CREATE TABLE statement [UltraLite]” [UltraLite - Database Management and Reference]
  ○ “CREATE INDEX statement [UltraLite]” [UltraLite - Database Management and Reference]
  ○ “CREATE PUBLICATION statement [UltraLite]” [UltraLite - Database Management and Reference]
  ○ “CREATE SYNCHRONIZATION PROFILE statement [UltraLite]” [UltraLite - Database Management and Reference]

● **CREATE SYNCHRONIZATION PROFILE now also supports OR REPLACE clause**  The CREATE SYNCHRONIZATION PROFILE statement now also support the option to replace a table. See “CREATE SYNCHRONIZATION PROFILE statement [UltraLite]” [UltraLite - Database Management and Reference].

● **File transfer through MobiLink**  UltraLiteJ can upload and download files in the remote database through the MobiLink server.

  The desktop version of UltraLiteJ can download any type of file from MobiLink to the local file system, or upload any type of file in the local file system to MobiLink.

  The BlackBerry OS version of UltraLiteJ can download valid, unencrypted, non-obfuscated database files from MobiLink and store them in the object store, or upload these types of databases to MobiLink. It can download any type of file from MobiLink to the media card or flash storage and upload any type of file from flash and media cards to MobiLink.

  For more information about UltraLiteJ file transfers, see “FileTransfer interface [UltraLiteJ]” [UltraLite - Java Programming].

● **UltraLiteJ Database Transfer utility can open and transfer databases on the BlackBerry's file system**  UltraLiteJ automatically decides the location of the database (object store or file
system) based on the database name. If the name starts with file:// (case sensitive), then the utility will try to find the database in the file system; otherwise it will find the database in the object store.

- **BlackBerry install directory renamed**  The install directory for BlackBerry smartphone files has been renamed to use the minimum BlackBerry OS version. The UltraLite\UltraLiteJ\Java MERIM11 directory is now UltraLite\UltraLiteJ\BlackBerry4.2 to indicate files compatible with BlackBerry OS 4.2 and later.

- **Blobfile type support for ULjLoad and ULjUnload**  The UltraLiteJ load and unload utilities support custom implementations of the blobfile type. For a sample UltraLiteJ database implementation of blobfile types, see “UltraLite Java Edition Database Unload utility (uljunload)” [UltraLite - Database Management and Reference] and “UltraLite Java Edition Database Load utility (uljload)” [UltraLite - Database Management and Reference].

- **Improved row limiting algorithm**  The row limiting algorithm has been improved to consider that rows from tables with many columns may use more resources than rows from tables with few columns.

- **System table changes**  The following changes have been made to the UltraLiteJ system tables:
  - column_default_value column in syscolumn system table supports the DEFAULT AUTOFIENAME default clause  This column can handle VARCHAR types that are specified with a column default value of DEFAULT AUTOFIENAME. See “syscolumn system table” [UltraLite - Database Management and Reference].
  - filename_colid column added to syscolumn system table  This column stores the column id of the referenced file_name column in the schema definition; otherwise, this column is null. See “syscolumn system table” [UltraLite - Database Management and Reference].
  - table_partition_size column added to systable system table  This column stores the defined partition size value. See “systable system table” [UltraLite - Database Management and Reference].

- **Encryption performance changes**  The following improvements have been made to the EncryptionControl interface to improve performance in slow CPU environments:
  - UltraLiteJ now only encrypts data and system critical pages.
  - The decrypt method now accepts an additional parameter to specify the number of bytes that need to be decrypted.

**New UltraLiteJ API features**

- **API features**  The following objects have been added to this release:
  - The ConfigFileME interface has been added to provide methods that allow configurations for persistent databases saved in a file on a Java ME device file system.
  - The setRowScoreFlushSize and setRowScoreMaximum methods in the ConfigPersistent interface have been added to replace the setRowMinimumThreshold and setRowMaximumThreshold methods. These methods accommodate the new row limiting algorithm. The row limiting algorithm
improves the resource management of tables with many columns. See “ConfigPersistent.setRowScoreFlushSize method [BlackBerry] [UltraLiteJ]” [UltraLite - Java Programming] and “ConfigPersistent.setRowScoreMaximum method [BlackBerry] [UltraLiteJ]” [UltraLite - Java Programming].

The UUIDValue interface has been added to describe a unique identifier. See “UUIDValue interface [UltraLiteJ]” [UltraLite - Java Programming].

○ The getLastIdentity method in the Connection interface has been added to retrieve the most recent value inserted into a DEFAULT AUTOINCREMENT or DEFAULT GLOBAL AUTOINCREMENT column through the current connection. See “Connection.getLastIdentity method [UltraLiteJ]” [UltraLite - Java Programming].

○ The getSyncObserver method in the Connection interface has been added to return the currently registered SyncObserver object for the connection. See “Connection.getSyncObserver method [UltraLiteJ]” [UltraLite - Java Programming].

○ The getSyncResult method in the Connection interface has been added to return the result of the last SYNCHRONIZE SQL statement executed on the connection. See “Connection.getSyncResult method [UltraLiteJ]” [UltraLite - Java Programming].

○ The setSyncObserver method in the Connection interface has been added to set a SyncObserver object to monitor the progress of synchronizations on the connection. See “Connection.setSyncObserver method [UltraLiteJ]” [UltraLite - Java Programming].

○ The createFileTransfer and createObjectStoreTransfer methods in the DatabaseManager interface have been added to create FileTransfer objects that can be used to transfer database files. See “DatabaseManager.createFileTransfer method [UltraLiteJ]” [UltraLite - Java Programming] and “DatabaseManager.createObjectStoreTransfer method [BlackBerry] [UltraLiteJ]” [UltraLite - Java Programming].

○ The FileTransfer interface has been added to provide methods that specify the options for a file transfer. See “FileTransfer interface [UltraLiteJ]” [UltraLite - Java Programming].

○ The FileTransferProgressData interface has been added to allow data to be passed to the observer callback. See “FileTransferProgressData interface [UltraLiteJ]” [UltraLite - Java Programming].

○ The FileTransferProgressListener interface has been added as the file transfer observer. See “FileTransferProgressListener interface [UltraLiteJ]” [UltraLite - Java Programming].

○ The getPlanTree method in the PreparedStatement interface has been added to present the plan for a query in a more readable fashion when displayed on a computer monitor or when printed. See “PreparedStatement.getPlanTree method [UltraLiteJ]” [UltraLite - Java Programming].

○ The hasShadowPaging method in the ConfigPersistent interface has been added to detect if shadow paging has been turned on. See “ConfigPersistent.hasShadowPaging method [BlackBerry] [UltraLiteJ]” [UltraLite - Java Programming].
UltraLiteJ behavior changes and deprecated features

Following is a list of deprecated features and behavior changes to UltraLiteJ introduced in version 12.0.0. For information about supported platforms and versions, see http://www.sybase.com/detail?id=1061806.

Removed, deprecated, and modified utility options
- **-p** option for **ULjInfo**, **ULjLoad** and **ULjUnload** is now optional. **dba** is used as the default password when the **-p** option is not specified.

Depreciated platforms
- UltraLiteJ no longer supports BlackBerry OS 4.1. UltraLiteJ now supports BlackBerry OS 4.2 or later.

Removed, deprecated, and modified APIs
- **Modified UltraLiteJ API objects**
  - The `anywhere.ultralitej` package name has changed. The package name now include the version number to allow multiple versions of UltraLiteJ to co-exist in the BlackBerry and Java ME environment, UltraLiteJ JAR files, COD files. See “UltraLiteJ API reference” [UltraLite - Java Programming].
  - The `table_autoinc` column in UltraLiteJ `syscolumn` system table has been removed. See “syscolumn system table” [UltraLite - Database Management and Reference].
  - The `CollectionOfValueReaders` interface has been removed. All `CollectionOfValueReader` methods have been moved to the `ResultSet` interface. See “ResultSet interface [UltraLiteJ]” [UltraLite - Java Programming].
  - The `CollectionOfValueWriters` interface has been removed. All `CollectionOfValueWriter` methods have been moved to the `PreparedStatement` interface. See “PreparedStatement interface [UltraLiteJ]” [UltraLite - Java Programming].
  - The `ConfigPersistent.setRowMaximumThreshold` and `setRowMinimumThreshold` have been replaced by `setRowScoreFlushSize` and `setRowScoreMaximum`. See “ConfigPersistent.setRowScoreFlushSize method [BlackBerry] [UltraLiteJ]” [UltraLite - Java Programming] and “ConfigPersistent.setRowScoreMaximum method [BlackBerry] [UltraLiteJ]” [UltraLite - Java Programming].
All schema-related methods and interfaces have been removed. The ForeignKeySchema interface and all methods in the ColumnSchema, IndexSchema, and TableSchema interfaces have been removed. The following methods have been removed from the Connection interface:

- createPublication
- createTable
- dropForeignKey
- dropPublication
- dropTable
- enableSynchronization
- renameTable
- schemaCreateBegin
- schemaCreateComplete
- setNeverSynchronized
- startSynchronizationDelete
- stopSynchronizationDelete
- truncateTable

Use UltraLiteJ supported SQL statements, such as CREATE TABLE and CREATE PUBLICATION, to perform schema operations. See “UltraLite SQL statements” [UltraLite - Database Management and Reference].

The setDatabaseId and getDatabasePartitionSize methods of the Connection interface have been removed. A default partition size can no longer be specified. Use the DEFAULT GLOBAL AUTOINCREMENT statement to override the default partition size. See “GLOBAL AUTOINCREMENT columns in UltraLite” [UltraLite - Database Management and Reference].

The getDatabaseID method of the Connection interface now has the same effect as calling Connection.getOption(OPTION_DATABASE_ID). See “Connection.getOption method [BlackBerry] [UltraLiteJ]” [UltraLite - Java Programming].

The Value, ValueReader, and ValueWriter classes have been removed.

Miscellaneous
- **User publication limit increase**  The maximum number of user publications has increased to 63.

**Administration tools new features**

Following is a list of additions to administration tools introduced in version 12.0.0. For information about supported platforms and versions, see [http://www.sybase.com/detail?id=1061806](http://www.sybase.com/detail?id=1061806).

- **Connect window**  The layout of the Connect window has been streamlined in version 12.0.0 and the Connect Assistant has been removed. Previously, you had to know which options were required for the type of connection you wanted to make. Now, you choose the connection type and the Connect window presents you with the options that are applicable to your specified connection type.
Caution

○ You must choose the one of the following connection types from the Action dropdown list:

**Connect To A Running Database On This Computer**  Connects to a database that is already running on your computer.

**Connect With An ODBC Data Source**  Connect to a database using an ODBC data source.

**Connect To A Running Database On Another Computer**  Connects to a database that is running on another computer in the network.

**Start And Connect To A Database On This Computer**  Starts a database on this computer and connects to it.

**Start And Connect To A Database On Another Computer**  Starts a database over a network on another computer and connects to it.

**Connect With A Connection String**  Connects to a database using a connection string.

The options below the Action dropdown list change depending upon your choice.

○ In Interactive SQL, underneath the Connect To A SQL Anywhere Database heading, click Change Database Type to change the type of database that you are connecting to.

For example, when you are connecting to a database with an ODBC data source, the Connect window displays only two options: ODBC Data Source Name and ODBC Data Source File.

If required, you can click Advanced to specify options such as TCP/IP, encryption, and other advanced options. See “Opening the Connect window (Sybase Central)” [SQL Anywhere Server - Database Administration].

- **New default for checking software updates**  Now, Interactive SQL, Sybase Central, SQL Anywhere Console utility, and MobiLink Monitor check for updates daily by default. In previous versions of the software, the default was to never check. See “Software updates” [SQL Anywhere Server - Database Administration].

- **Viewing images, HTML, and XML data in Interactive SQL and Sybase Central**  You can preview images, HTML, and XML data in a result set. See “Viewing images and SVGs in Interactive SQL” [SQL Anywhere Server - Database Administration] and “Viewing HTML and XML data in Interactive SQL” [SQL Anywhere Server - Database Administration].

- **New system tray icons for Sybase Central and Interactive SQL**  When the Sybase Central or the Interactive SQL fast launcher option is enabled, an icon now appears in the system tray. Right-click the icon and click Open to open the application or click Exit to close the application (if it is running) and terminate the fast launcher process. See “Fast launcher option” [SQL Anywhere Server - Database Administration].

- **Accessibility Enablement option**  The Accessibility Enablement option is now installed by default. Previously you had to install this option separately. The Accessibility Enablement option
allows Interactive SQL, Sybase Central, the SQL Anywhere Console utility, and the MobiLink Monitor to work with accessibility aids such as screen readers. See “Accessibility using the Java Access Bridge” [SQL Anywhere 16 - Introduction].

**Sybase Central plug-in new features**

Following is a list of additions to Sybase Central plug-ins introduced in version 12.0.0.

### SQL Anywhere plug-in new features

- **View the SQL statements and utility commands created by wizards**  
  Most SQL Anywhere plug-in wizards that create a database object or run a database utility include a new page at the end of the wizard. This page displays the SQL statements and utility commands that are executed when you click **Finish**. Clicking **Finish** executes the SQL statements and/or runs the utility commands.

  Alternatively, you can copy the SQL statement to the clipboard, click **Cancel** to exit the wizard, and then execute the SQL statements via Interactive SQL. This feature allows you to use the wizards to generate SQL scripts without modifying the database. See “Viewing SQL statements and utility commands generated by wizards” [SQL Anywhere Server - Database Administration].

- **Support for spatial data**  
  For information about the new plug-in features, including wizards, related to spatial data, see “Support for spatial data” on page 115.

- **New database Fragmentation tab**  
  You can view a graphical representation of the fragmentation of base tables and indexes. The **Fragmentation** tab provides a graphical representation of the results from running sa_table_fragmentation system procedure on base tables. See “The Fragmentation tab (SQL Anywhere plug-in)” [SQL Anywhere Server - SQL Usage].

- **Logging database changes**  
  You can log all SQL statements executed by the SQL Anywhere plug-in that modify the database. The log is stored in a .sql file. See “Logging SQL statements” [SQL Anywhere Server - Database Administration].

- **Support for WITH NULLS NOT DISTINCT clause, CREATE INDEX statement**  
  You can use the SQL Anywhere plug-in to, create, view, and alter indexes that use the WITH NULLS NOT DISTINCT clause. The Indexes folder contains a **Nulls Distinct** column. The value in the column is blank when the index is a primary key, foreign key, unique constraint, or a non-unique index.

  See the UNIQUE clause of the “CREATE INDEX statement” [SQL Anywhere Server - SQL Reference].

- **Term breakers**  
  The SQL Anywhere plug-in supports external term breakers and prefilters for text configuration objects. See “ALTER TEXT CONFIGURATION statement” [SQL Anywhere Server - SQL Reference].

- **Improvements to database documentation**  
  When you use the **Database Documentation Wizard**, the generated documentation contains information about tables. The table information includes the owner name, procedures that modify the table, column information, and comments. See “Documenting a database” [SQL Anywhere Server - Database Administration].
• **Improvements to the Configure Type Filter window**  The Configure Type Filter window lets you specify which database objects appear in the folder list in the left pane of Sybase Central. You can set your specifications to be the default type filter. Sybase Central uses the default filter whenever it connects to a database that does not have a type filter specified. See “Sybase Central navigation” [SQL Anywhere Server - Database Administration].

• **Improvements to Create Maintenance Plan Wizard**  Maintenance plans contain user-defined operations. In the Create Maintenance Plan Wizard you can add user-defined operations as SQL statements that execute either before validation or after backup.

In addition, the Maintenance Plans folder now includes a Status column. See “Maintenance plans” [SQL Anywhere Server - Database Administration].

• **Improvements to the Create Function Wizard**  Previously, the Create Function Wizard limited the return type to a built-in type. Now, you can choose either a built-in type or a domain within the wizard. See “Creating a user-defined function” [SQL Anywhere Server - SQL Usage].

• **Improvements to the Backup Database Wizard**  By default the Backup Database Wizard now enables free page elimination. See “Creating an archive backup (Sybase Central)” [SQL Anywhere Server - Database Administration].

• **Overview tab enhancement**  The database Overview tab shows information about the health and statistics of copy servers along with information about database mirroring. See “Database health and statistics” [SQL Anywhere Server - Database Administration].

• **Improvements to the Synchronization Subscription Properties window**  For databases created with SQL Anywhere 12 and later, the name of the synchronization subscription appears in the Synchronization Subscription Properties window, as well as on the Synchronization Subscriptions tab. For databases that are created with earlier versions of SQL Anywhere, the name appears as (unnamed). See “Synchronization subscription creation” [MobiLink - Client Administration].

• **New Synchronize Using Synchronization Profile window**  You can synchronize a SQL Anywhere database by right-clicking a synchronization profile, clicking Synchronize, and clicking OK. See “MobiLink synchronization profiles” [MobiLink - Client Administration].

**Sybase Central behavior changes and deprecated features**

Following is a list of changes to Sybase Central introduced in version 12.0.0.

• **Sybase Central version changes to 6.1.0**  SQL Anywhere 12 includes version 6.1.0 of Sybase Central 6.1.0.

• **Sybase Central only supports SQL Anywhere versions 10x and later databases**  Support for version 9 database servers and databases created with version 9 software has been removed from the SQL Anywhere plug-in. When unloading and reloading the database into a reload file, or into a new or existing database, you can still connect to a database created with version 5, 6, 7, 8 or 9 software running on a version 9 or later database server.
For databases created by version 5, 6, 7, 8, or 9 software running on a version 6, 7, 8, or 9 database server, you can connect to the database from the SQL Anywhere plug-in temporarily to do one of the following tasks:

- Unload the database into a reload file.
- Unload and reload the database into a new version 12 database.
- Unload and reload the database into an existing version 12 database.

The files for the database being unloaded must be located on the local computer.

There is no support in the SQL Anywhere plug-in for databases created by version 4 or earlier software running on a version 5 or earlier server.

See “SQL Anywhere Server upgrades” on page 272.

- **Support for 32- and 64-bit computers**  The Sybase Central configuration file has been renamed:

  - On 32-bit computers, the `.scRepository600` file is now named `.scRepository610_32`

  - On 64-bit computers, the `.scRepository600` file can be named `.scRepository610_32` and/or `.scRepository610_64` depending upon your installation.

- **When deploying Sybase Central, you no longer need to create JPR files for the plug-ins**  Previously, when deploying Sybase Central and SQL Anywhere you needed to create JPR files for each plug-in. Now, Sybase Central uses the environment variable `%SQLANY12%` to find and register the plug-ins.

### SQL Anywhere plug-in changed features

- **Maintenance plan enhancements**  The following enhancements have been made to maintenance plans:

  - You can include SQL statements that are executed before validation
  - You can include SQL statements that are executed after backup
  - You can view the status of a maintenance plan while it is running
  - You can only run maintenance plan at a time

  For more information, see “Maintenance plans” [SQL Anywhere Server - Database Administration].

- **Automatically refresh dynamic objects and properties**  The lists of connections and locks, as well as the dynamic properties for events, maintenance plans, and windows services are automatically refreshed every 10 seconds by default. See “Setting the refresh frequency” [SQL Anywhere Server - Database Administration].

- **New default for the Create Database Wizard**  When you use the Create Database Wizard to create a new version 12 database, global checksums are now enabled by default. When you use the wizard to create a version 11 or later database, global checksums are disabled by default. Global checksums is always enabled by default when creating databases for Windows Mobile. See “Checksums enabled by default for new databases” on page 140.
Interactive SQL new features

Following is a list of additions to Interactive SQL introduced in version 12.0.0.

- **View spatial data using the Spatial Preview and Spatial Viewer** See “Viewing spatial data as images (Interactive SQL)” [SQL Anywhere Server - Spatial Data Support].

- **New ways to execute COMMIT and ROLLBACK statements** In Interactive SQL you can click SQL » Commit to execute a COMMIT statement and you can click SQL » Rollback to execute a ROLLBACK statement. You can also use the keyboard shortcuts: Ctrl+Shift+C to execute a COMMIT statement and Ctrl+Shift+R to execute a ROLLBACK statement.

  Executing a COMMIT or ROLLBACK via the SQL menu or a keyboard shortcut does not modify the contents of the SQL Statements pane; however, the Results tab in the Results pane is cleared. See “Keyboard shortcuts for Interactive SQL” [SQL Anywhere Server - Database Administration].

- **Changes to how you can select and copy columns, rows, and cells from result sets** In the Interactive SQL Results pane you can select multiple columns, rows, and cells in a result set, and then copy them. For example, to select multiple columns, hold the Ctrl key while clicking cells from the columns you want to copy, and then right-click and choose Copy Data » Columns. See “Copying rows, columns, and cells from an Interactive SQL result set” [SQL Anywhere Server - Database Administration].

- **Prevent OEM users from saving passwords in their favorites** OEM deployments can now prevent users from saving passwords in their connection favorites in Interactive SQL. See the allowPasswordsInFavorites option in “Administration tools configuration” [SQL Anywhere Server - Programming].

- **Editing, importing, and exporting Favorites** Now you can edit, export, and import your Interactive SQL favorites. See “Adding SQL script files, SQL statements, and connections to favorites” [SQL Anywhere Server - Database Administration] and “Sharing favorites” [SQL Anywhere Server - Database Administration].

- **Execution times** The status bar in Interactive SQL shows the length of time that the current SQL statement has been executing.

- **Changes to the Results pane**
  - **Display results as text or in a scrolling table** Previously, you could only configure the display of a result set in the Results pane by changing settings in the Options window. Now, you can click Data » Show Results As Scrollable Table to display the result set in a scrollable table. You can also click Data » Show Results As Text to display the result set as text. You must execute a statement to see the change take effect. See “Customizing Interactive SQL” [SQL Anywhere Server - Database Administration].
  - **Sizing columns in Interactive SQL** You can right-click a result set and choose whether the columns should be sized to fit the window or to fit the data. See “Customizing Interactive SQL” [SQL Anywhere Server - Database Administration].
● **Help for SQL functions**  In the SQL Statements pane, you can right-click a SQL function and click Help For function-name and the documentation for the function appears.

● **Suppress warning messages in Interactive SQL**  You can disable some of the warning messages that appear in Interactive SQL. For example you can suppress the warning that appears when you have unsaved text in the SQL Statements pane.

You can disable the warning message by clicking Tools » Options » Messages and then clearing the checkboxes in the Optional Messages list. See “Customizing Interactive SQL” [SQL Anywhere Server - Database Administration].

● **Recovering files in Interactive SQL**  Interactive SQL attempts to recover unsaved changes to .sql files when Interactive SQL closes unexpectedly. When you edit a file, Interactive SQL makes a backup copy of the file 30 seconds after the last change and before you execute a statement.

**Interactive SQL behavior changes and deprecated features**

Following is a list of changes to Interactive SQL introduced in version 12.0.0.

● **Support for 32- and 64-bit computers**

  ○ Now, you can have both 32- and 64-bit versions of Interactive SQL installed on the same computer.

● **Changes to the default encoding for Windows**  The following change applies to running Interactive SQL as a console application (with no windowed user interface) on Windows computers where the ANSI and OEM encodings differ, for example on a U.S. English Windows XP computer.

1. Previously, when running Interactive SQL as a console application, the INPUT and READ statements assumed that the file was encoded using the OEM encoding (cp437 on a U.S. English Windows XP computer) in the absence of an explicit ENCODING clause. Similarly, the OUTPUT statement would output the file using the OEM encoding.

   Now, when running Interactive SQL as a console application, the INPUT and READ statements assume that the file is encoded using the ANSI encoding (cp1252 on U.S. English Windows XP computer). Similarly, the OUTPUT statement outputs the file using the ANSI encoding.

   At a command prompt, to process a file that uses the OEM encoding, you must specify the encoding explicitly. For example:

   ```
   dbisql READ ENCODING 'cp437' myfile.sql
   ```

2. Previously, when running Interactive SQL as a console application, results written to and read at a command prompt used the ANSI encoding (cp1252 on a U.S. Windows XP computer), which could cause extended characters to be displayed incorrectly.

   Now, when running Interactive SQL as a console application, results written to and read at a command prompt use the OEM encoding (cp437 on a U.S. Windows XP computer).

See “default_isql_encoding option [Interactive SQL]” [SQL Anywhere Server - Database Administration].
**Change to the CLEAR statement, Clear menu item, and the Esc key**  
Now, the CLEAR statement closes any open result sets and leaves the contents of the SQL Statements pane unchanged. See “CLEAR statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference].

Also, the Edit » Close Results menu item, which is equivalent to executing a CLEAR statement, closes any open result sets and leaves the contents of the SQL Statements pane unchanged.

The Edit » Clear menu item which used to clear the contents of the SQL Statements pane has been removed. As a result, the keyboard shortcut for the Clear menu item, the Esc key, has also been removed. Now by default, pressing the Esc key has no affect.

However, you can set the Esc key to clear the SQL Statements pane and close any opened result sets. Click Tools » Options » Compatibility and click Pressing The Esc Key Clears SQL Statements And Closes Result Sets.

**-codepage option removed**  
If you want Interactive SQL to read a file with a specific code page, use the ENCODING clause of the INPUT, OUTPUT, or READ statements. The -codepage option has been removed from the software. See:

- “Interactive SQL utility (dbisql)” [SQL Anywhere Server - Database Administration]
- “Interactive SQL for UltraLite utility (dbisql)” [UltraLite - Database Management and Reference]
- “INPUT statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference]
- “OUTPUT statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference]
- “READ statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference]

**Change to SET OPTION statement [Interactive SQL]**  
Previously, if you used the SET OPTION statement to set an option and didn’t specify a value, the option was set to Off. Now, if the option value is omitted, the specified option is set to its default value. This change affects the following options: auto_commit, auto_refetch, bell, commit_on_exit, and echo. See “SET OPTION statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference].

**Change to OUTPUT statement**  
When outputting results to a TEXT file, you can use the WITH COLUMN NAMES clause to insert the column names at the beginning of the file. See “OUTPUT statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference].

**Change to INPUT statement**  
When inserting lines from a TEXT file with the INPUT statement, you can now use the SKIP clause to specify a number of lines to omit from the start of the file. See “INPUT statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference].

**READ keyword no longer required**  
When you run Interactive SQL at a command prompt, the READ keyword is optional when specifying a .sql file to run. If the .sql file requires parameters, specify the parameters after the file name.

For example, the following two commands are equivalent:

<table>
<thead>
<tr>
<th>With the READ keyword</th>
<th>Without the READ keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ  file.sql parm1</td>
<td>file.sql parm1</td>
</tr>
</tbody>
</table>

See “Interactive SQL utility (dbisql)” [SQL Anywhere Server - Database Administration].
● **Improvements to Microsoft Excel ODBC driver support**  The following list describes changes related to exporting data from SQL Anywhere to Excel files via the Microsoft Excel ODBC driver:

○ Previously you could not export data that was stored as a CHAR, LONG VARCHARE, NCHAR, NVARCHAR, or LONG NVARCHAR data types.

   Now when you export data that is stored as a CHAR, LONG VARCHARE, NCHAR, NVARCHAR, or LONG NVARCHAR data types from a SQL Anywhere database using the Microsoft Excel Anywhere ODBC driver, the data is stored as VARCHAR (the closest type supported by the Excel driver).

   The Microsoft Excel ODBC driver supports text column widths up to 255 characters.

○ You can export data that is stored as REAL, FLOAT, and BIGINT data types.

○ Data stored as MONEY and SMALLENUMBER data types are exported to the CURRENCY data type. Otherwise numerical data is exported as numbers.

○ You can use the [Export Wizard](#) to export tables.

See “Data export” [*SQL Anywhere Server - SQL Usage*].

---

### SQL Anywhere Monitor new features

Following is a list of additions to SQL Anywhere Monitor introduced in version 12.0.0.

● **New user interface**  The Monitor user interface is dashboard-based. Dashboards contain widgets to display metrics, alerts, and resource information. Dashboards are specific to each user. Any user can add, edit, or delete their dashboards. See “Dashboards” [*SQL Anywhere Server - Database Administration*].

● **Monitor MobiLink server farms and Relay Server farms**  You can use the Monitor to monitor MobiLink server farms and Relay Server Farms, as well as SQL Anywhere databases and MobiLink servers. See “SQL Anywhere Monitor” [*SQL Anywhere Server - Database Administration*].

● **Close connections from within the Monitor**  From within the Monitor, you can close connections to resource databases. See “Closing a connection using the Monitor” [*SQL Anywhere Server - Database Administration*].

● **Import SQL Anywhere resources to be monitored**  You can create a CSV file that contains a list of resources, and import this list into the Monitor. Previously, you could only add one resource at a time to the Monitor. See “Adding multiple resources with a CSV file” [*SQL Anywhere Server - Database Administration*].

● **Perform on-demand maintenance**  Administrators can perform unscheduled maintenance on the Monitor. See “Backing up the Monitor” [*SQL Anywhere Server - Database Administration*].

● **New backup alert for SQL Anywhere resources**  Administrators can configure the Monitor to issue an alert when a SQL Anywhere resource has not been successfully backed up for a given number of days. By default, the Monitor issues an alert after 14 days have past since a resource was successfully backed up. See “Specifying alert thresholds” [*SQL Anywhere Server - Database Administration*].
Time is reported local to the browser  The times reported in the Monitor are always local to the browser you are using. To calculate the difference between the browser time and the resource time, see “Understanding how time is displayed” [SQL Anywhere Server - Database Administration].

Export metrics  You can export metrics that have a graph or table associated with them to an XML file. For example, most of the metrics in the Key Performance Metrics widget can be exported. See “Exporting metrics” [SQL Anywhere Server - Database Administration].

Troubleshooting features  When you need to troubleshoot the Monitor, Administrators can use the Message Log and Exception Reports features. See “Viewing the Message Log” [SQL Anywhere Server - Database Administration] and “Viewing exception reports” [SQL Anywhere Server - Database Administration].

SQL Anywhere Monitor behavior changes

Following is a list of changes to SQL Anywhere Monitor introduced in version 12.0.0.

Changes to metric collection  Previously you could configure the type of metrics that the Monitor collected and set the alert thresholds. Now you can only configure the alert thresholds. See “SQL Anywhere Monitor” [SQL Anywhere Server - Database Administration].

Removing resources  You can remove a monitored resource from the Monitor without stopping it. Previously you had to stop the resource before you could remove it. See “Removing resources” [SQL Anywhere Server - Database Administration].

Resources no longer have states  Now resources only have statuses. See “Dashboards” [SQL Anywhere Server - Database Administration].

Changes to alert status  When a condition that triggered an alert no longer exists, the alert status changes to Inactive. An alert with the status Inactive indicates that the condition that triggered the alert is no longer present, but a Monitor user has not manually resolved it. In addition, the Monitor assigns a severity to each alert that it issues. See “Dashboards” [SQL Anywhere Server - Database Administration].

Changes to the alert threshold defaults  Previously, the Monitor issued alerts when the CPU memory usage reached 85 percent of the maximum cache size for two contiguous collection intervals. Now, the default is 90 percent and the time threshold is 30 seconds. See “Alert definitions and thresholds” [SQL Anywhere Server - Database Administration].

All users are required to log in into the Monitor  Previously, the Monitor did not require anyone to log in to have read-only access. Now all users are required to log in to the Monitor. See “Monitor users” [SQL Anywhere Server - Database Administration].

Changes to user language preferences  When a user logs in, the Monitor uses the language preferences of that user to configure the language displayed in the Monitor and used in alerts. Previously, the Monitor used the language set in the browser. See “Editing a Monitor user” [SQL Anywhere Server - Database Administration].
Documentation enhancements

Following is a list of changes made to the SQL Anywhere documentation in version 12.0.0.

- **Frequently Asked Questions (FAQ)** A comprehensive list of frequently asked questions is now available. Intended primarily for new and intermediate users, the FAQ attempts to answer many of the common questions that are asked regularly in forum discussions. See “Frequently asked questions - SQL Anywhere” [SQL Anywhere 16 - Introduction].

- **Database server performance warnings** Descriptions are now provided for the performance warnings that appear in the database server messages window, as well as links to information about how to address the performance issue. See “Troubleshooting database server performance warnings” [SQL Anywhere Server - Database Administration].

- **DocCommentXchange is the default documentation** DocCommentXchange is the default documentation format for SQL Anywhere 12. DocCommentXchange is an online community for accessing and discussing SQL Anywhere documentation on the web.

- **New Developer Centers on Sybase.com** To supplement the SQL Anywhere documentation, consider visiting the SQL Anywhere Developer Centers at http://www.sybase.com/developer/library/sql-anywhere-techcorner to browse technical white papers, FAQs, tech notes, downloads, techcasts and more for answers to your questions.
What's new in version 11.0.1

Note
As with all forward-looking statements, the lists of deprecated features are not guaranteed to be complete and are subject to change.

For information about new features and behavior changes in versions of SQL Anywhere before version 10, see http://dcx.sybase.com/html/dbwnen10/dbwnen10.html.

SQL Anywhere new features

Following is a list of additions to SQL Anywhere databases and database servers introduced in version 11.0.1.

Connection properties

The following connection properties have been added in SQL Anywhere version 11.0.1:

- Authenticated
- IsDebugger
- QueryBypassedCosted
- QueryBypassedHeuristic
- QueryBypassedOptimized
- QueryOpened
- QueryDescribedBypass
- QueryDescribedOptimizer
- StatementDescribes
- StatementPostAnnotates
- StatementPostAnnotatesSimple
- StatementPostAnnotatesSkipped
- WaitStartTime
- WaitType

For descriptions of these properties, see “List of connection properties” [SQL Anywhere Server - Database Administration].

Database properties

The following database properties have been added in SQL Anywhere version 11.0.1:
Database server properties

Following is a list of enhancements made to database server properties in SQL Anywhere version 11.0.1.

- ServerEdition

For descriptions of these properties, see “List of database server properties” [SQL Anywhere Server - Database Administration].

Database utilities

Following is a list of enhancements made to database utilities in SQL Anywhere version 11.0.1.

- Service utility (dbsvc) enhancements
  On Windows, you can now create services for the MobiLink Relay Server (rhost), relay server outbound enabler (RSOLE), and Volume Shadow Copy Service (dbvss11) using the Service utility. See “Service utility (dbsvc) for Windows” [SQL Anywhere Server - Database Administration].

- Unload utility (dbunload) enhancements
  The dbunload utility now supports the following options:
  - *-cm option* Displays the dbinit command or CREATE DATABASE statement for the database being unloaded.
  - *-l option* Retains the next available value for AUTOINCREMENT columns in the rebuilt database.

  See “Unload utility (dbunload)” [SQL Anywhere Server - Database Administration].
System procedures and functions

Following is a list of system procedure and function enhancements added in SQL Anywhere version 11.0.1.

- **sa_get_table_definition system procedure**  The new `sa_get_table_definition` system procedure returns a `LONG VARCHAR` string containing the SQL statements required to create the specified table and its indexes, foreign keys, triggers and granted permissions. See “`sa_get_table_definition` system procedure” [SQL Anywhere Server - SQL Reference].

- **FIRST_VALUE function [Aggregate]**  The `FIRST_VALUE` function [Aggregate] now includes a RESPECT NULLS clause. See “`FIRST_VALUE function [Aggregate]`” [SQL Anywhere Server - SQL Reference].

- **LAST_VALUE function [Aggregate]**  The `LAST_VALUE` function [Aggregate] now includes a RESPECT NULLS clause. See “`LAST_VALUE function [Aggregate]`” [SQL Anywhere Server - SQL Reference].

- **sa_set_http_option system procedure**  The `AcceptCharset` option now allows more control over character set selection. See “`sa_set_http_option system procedure`” [SQL Anywhere Server - SQL Reference].

SQL statements

Following is a list of SQL enhancements introduced in SQL Anywhere version 11.0.1.

- **New DEFAULT VALUES clause, INSERT statement**  The new DEFAULT VALUES clause of the INSERT statement allows you to insert the default values for all columns. See “`INSERT statement`” [SQL Anywhere Server - SQL Reference].

- **CREATE ENCRYPTED DATABASE statement**  This statement creates an encrypted copy of an existing database, including all transaction logs, mirror logs, and dbspaces. You can also use this statement to create a copy of a database with table encryption enabled. See “`CREATE ENCRYPTED DATABASE statement`” [SQL Anywhere Server - SQL Reference].

  If you want to encrypt a database that requires recovery, for example to send to Technical Support, you must still use the CREATE ENCRYPTED FILE statement. See “`CREATE ENCRYPTED FILE statement`” [SQL Anywhere Server - SQL Reference].

- **CREATE DECRYPTED DATABASE statement**  This statement creates a decrypted copy of an existing database, including all transaction logs, mirror logs, and dbspaces. See “`CREATE DECRYPTED DATABASE statement`” [SQL Anywhere Server - SQL Reference].

  If you want to decrypt a database that requires recovery, for example to send to Technical Support, you must still use the CREATE DECRYPTED FILE statement. See “`CREATE DECRYPTED FILE statement`” [SQL Anywhere Server - SQL Reference].
ALTER DATABASE statement enhancement  Attempting to execute an ALTER DATABASE UPGRADE statement on a database server that is currently being mirrored now results in an error. See “ALTER DATABASE statement” [SQL Anywhere Server - SQL Reference].

MESSAGE statement enhancement  The IMMEDIATE clause causes the message to be received by the client's message callback routine within a short period of time, regardless of whether the connection is idle or making requests. See “MESSAGE statement” [SQL Anywhere Server - SQL Reference].

Create or replace a function, procedure, trigger, or view of the same name  The new OR REPLACE clause allows you to create or replace a function, procedure, trigger, or view of the same name. See:

- “CREATE FUNCTION statement [Web service]” [SQL Anywhere Server - SQL Reference]
- “CREATE PROCEDURE statement [Web service]” [SQL Anywhere Server - SQL Reference]
- “CREATE TRIGGER statement” [SQL Anywhere Server - SQL Reference]
- “CREATE VIEW statement” [SQL Anywhere Server - SQL Reference]

Suppressing errors when a statement attempts to remove a database object that does not exist  The new IF EXISTS clause allows you to specify that you do not want an error returned when the DROP statement attempts to remove a database object that does not exist. See:

- “DROP EVENT statement” [SQL Anywhere Server - SQL Reference]
- “DROP FUNCTION statement” [SQL Anywhere Server - SQL Reference]
- “DROP MATERIALIZED VIEW statement” [SQL Anywhere Server - SQL Reference]
- “DROP PROCEDURE statement” [SQL Anywhere Server - SQL Reference]
- “DROP TABLE statement” [SQL Anywhere Server - SQL Reference]
- “DROP TRIGGER statement” [SQL Anywhere Server - SQL Reference]
- “DROP VIEW statement” [SQL Anywhere Server - SQL Reference]

New INTO LOCAL TEMPORARY TABLE clause, SELECT statement  The new INTO LOCAL TEMPORARY TABLE clause of the SELECT statement allows you to create and populate a local temporary table with the result set of a SELECT statement. Previously, you could only do this by using an INTO clause if the temporary table name started with #. See “SELECT statement” [SQL Anywhere Server - SQL Reference].

New IF NOT EXISTS clause, CREATE TABLE statement  The new IF NOT EXISTS clause of the CREATE TABLE statement allows you to create permanent, global temporary, and local temporary tables if the table does not already exist. See “CREATE TABLE statement” [SQL Anywhere Server - SQL Reference].

Specify an owner when creating temporary procedures or functions  The CREATE FUNCTION and CREATE PROCEDURE statements now allow you to optionally specify the owner for a temporary procedure or function. See:

- “CREATE FUNCTION statement [Web service]” [SQL Anywhere Server - SQL Reference]
- “CREATE PROCEDURE statement [Web service]” [SQL Anywhere Server - SQL Reference]
Programming interfaces

Following is a list of enhancements made to programming interfaces in SQL Anywhere version 11.0.1.

- **New ASP.NET providers** The following new ASP.NET providers mimic the functionality of the standard ASP.NET providers, but store their data in a SQL Anywhere database rather than in a SQL Server database:
  - **Memberships** Allows logging in and out, managing users and their password
  - **Roles** Allows assigning users to groups, allowing simple and easy permission management
  - **Profiles** Stores user variables
  - **Web Parts Personalization** Manages the storage of Web Parts data, allowing users to personalize their view
  - **Web Events** Works with Health Monitoring to store flushed web event information in the database

See “SQL Anywhere ASP.NET providers” [SQL Anywhere Server - Programming].

- **Support for Ruby** SQL Anywhere now supports the Ruby open source programming language. See “SQL Anywhere Ruby API support” [SQL Anywhere Server - Programming].

- **OLE DB now supports CATALOGS and SCHEMATA rowsets** The CATALOGS and SCHEMATA rowsets for OLE DB are now supported. Since SQL Anywhere does not support the notion of catalogs the SQL Anywhere OLE DB provider returns a result set for CATALOGS containing all currently started databases, and their locations, instead. Likewise, for SCHEMATA, the database name is used as the catalog in the result set.

- **Support added for ADO.NET Entity Framework** SQL Anywhere now provides support for the ADO.NET Entity Framework model. See “SQL Anywhere .NET Data Provider features” [SQL Anywhere Server - Programming].

Miscellaneous

Following is a list of miscellaneous enhancements introduced in SQL Anywhere version 11.0.1.

- **Improved performance for simple, inexpensive statement execution** The cost of generating an execution plan for statements with inexpensive execution times can sometimes be greater than the cost of executing the statement. With the following enhancements, SQL Anywhere now recognizes an expanded class of simple statements with inexpensive execution times and allows them to bypass the optimizer.

  For more information, see “Eligibility to skip query processing phases” [SQL Anywhere Server - SQL Usage].

  - **New FORCE NO OPTIMIZATION clause** Previously, all statements that required cost-based optimization were processed by the optimizer. With the addition of the FORCE NO
OPTIMIZATION clause, you can specify that you want the statement to bypass the optimizer. If the statement is too complex to process in this way—possibly due to the setting of database options or characteristics of the schema or query—the statement fails and the database server returns an error.

The following statements support the new FORCE NO OPTIMIZATION clause:

- “DELETE statement” [SQL Anywhere Server - SQL Reference]
- “INSERT statement” [SQL Anywhere Server - SQL Reference]
- “SELECT statement” [SQL Anywhere Server - SQL Reference]
- “UPDATE statement” [SQL Anywhere Server - SQL Reference]

○ New connection properties

  - QueryBypassedCosted
  - QueryBypassedHeuristic
  - QueryBypassedOptimized
  - QueryOpened
  - QueryDescribedBypass
  - QueryDescribedOptimizer
  - StatementDescribes
  - StatementPostAnnotates
  - StatementPostAnnotatesSimple
  - StatementPostAnnotatesSkipped

○ New database properties

  - Prepares
  - QueryBypassedCosted
  - QueryBypassedHeuristic
  - QueryBypassedOptimized
  - QueryOpened
  - QueryDescribedBypass
  - QueryDescribedOptimizer
  - StatementDescribes
  - StatementPostAnnotates
  - StatementPostAnnotatesSimple
  - StatementPostAnnotatesSkipped

○ New Optimization Method field The Optimization Method field has been added to the Optimizer Statistics section of the graphical plan. The field returns the execution strategy chosen by the query optimizer. See “Execution plan components” [SQL Anywhere Server - SQL Usage].

- Prevent a database server from becoming the default database server The -xd server option prevents a database server from listening on the default TCP port, and prevents the database server from becoming the default database server. See “-xd database server option” [SQL Anywhere Server - Database Administration].
Support for parallel archive backups

The SQL Anywhere database server now supports parallel backups for server-side archive backups. Parallel database backups take advantage of physical I/O to perform read and write operations in parallel, instead of sequentially, which improves performance.

Two new clauses have been added to the BACKUP DATABASE statement to support parallel archive backups:

- WITH CHECKPOINT LOG [ NO ] COPY
- MAX WRITE { n | AUTO }

Version 11.0.0 and earlier database servers cannot restore archive backups generated with version 11.0.1 database servers. Version 11.0.1 database servers can restore backups produced by older database servers.

See “BACKUP statement” [SQL Anywhere Server - SQL Reference].

Running Developer Edition and Evaluation database servers on Mac OS X

You can now automatically start Developer Edition and Evaluation Edition database servers from the administration tools on Mac OS X.

SQL Anywhere behavior changes and deprecated features

Following is a list of changes to SQL Anywhere databases and database servers introduced in version 11.0.1, grouped by category.

Behavior changes

- Full text search

  The following behavior changes have been made to support for full text search:

  - Operator precedence is now applied

    Previously, no precedence was applied to operators in a query string. Now, the following operator precedence is applied:

    - NEAR, FUZZY operators
    - AND NOT operator
    - AND operator
    - OR operator

    See “Operator precedence in a CONTAINS search condition” [SQL Anywhere Server - SQL Reference].

  - NEAR clause arguments must be terms or prefix terms

    When you perform a proximity search, the arguments to the NEAR clause must be terms or prefix terms. See “CONTAINS search condition” [SQL Anywhere Server - SQL Reference], and “Proximity search” [SQL Anywhere Server - SQL Usage].

  - Use of hyphen and AND NOT clause

    Within a phrase, a hyphen is treated as a term breaker, not a special character. Outside of a phrase, the treatment of a hyphen depends on the syntax
surrounding the hyphen. See “Allowed syntax for hyphen (-)” [SQL Anywhere Server - SQL Reference], and “Using the AND NOT operator in full text searches” [SQL Anywhere Server - SQL Usage].

○ **Use of asterisk and prefix searching**  When performing a prefix search, the asterisk must be appended to a term and followed immediately by a space, or by the end of query string, or by one of the allowed special characters. See “Allowed syntax for asterisk (*)” [SQL Anywhere Server - SQL Reference], and “Prefix search” [SQL Anywhere Server - SQL Usage].

○ **Creating a duplicate text index now returns an error**  You can no longer create duplicate text indexes. A text index is considered a duplicate if the following settings are identical to those of an existing text index:

- the base table being referenced
- the columns to be indexed (order does not matter)
- the settings for the configuration object used (TERM BREAKER, MINIMUM TERM LENGTH, MAXIMUM TERM LENGTH, STOPLIST, collation information)

Duplicate text indexes created before SQL Anywhere 11.0.1 can remain in the database and do not cause errors when started with a 11.0.1 database server. However, if a database that contains duplicate text indexes is reloaded on version 11.0.1 or later, an error is returned.

To identify duplicate text indexes in an existing database, execute the following query:

```sql
SELECT LIST(i.index_name)
FROM SYS.SYSIDX i
  JOIN SYS.SYSTEXTIDX t ON i.object_id = t.index_id AND t.sequence = 1
  JOIN SYS.SYSTEXTCONFIG f ON f.object_id = t.text_config
  JOIN (
    SELECT table_id, index_id, LIST(column_id, ', ') ORDER BY column_id) col_id
    FROM SYS.SYSIDXCOL
    GROUP BY table_id, index_id) x
ON x.table_id = i.table_id AND x.index_id = i.index_id
WHERE i.index_category = 4
GROUP BY i.table_id, f.term_breaker, f.min_term_length,
  f.max_term_length,
  f.collation, ISNULL(f.char_stoplist, '-'),
  ISNULL(f.nchar_stoplist, '-'), x.col_id
HAVING count(**) > 1
```

This query works only if the strings representing STOPLIST are exactly the same or if NO STOPLIST is specified. For example, stoplists 'a b c' and 'a - b c' are not considered the same stoplist by this query, but would be considered the same during a check for duplicates during text index creation.

- **Regular expressions**  Changes have been made to the behavior for the SIMILAR TO and REGEXP search conditions, and the REGEXP_SUBSTR function. The overall intention of the changes is to continue to have SIMILAR TO be consistent with the ANSI/SQL standard, while making the behavior of REGEXP and REGEXP_SUBSTR consistent with Perl.
○ **Database collations and matching**  Previously, REGEXP and REGEXP_SUBSTR determined if a literal or character class range in the pattern matched the string by using the collation equivalence and sort order. Now, REGEXP and REGEXP_SUBSTR use binary comparisons of code point values for matching and evaluation of ranges. The change was made to make the behavior consistent with Perl 5.0.

SIMILAR TO still uses the database collation for matching and range evaluation. See “LIKE, REGEXP, and SIMILAR TO search conditions” [SQL Anywhere Server - SQL Reference].

○ **Database case sensitivity and [:upper:] and [:lower:] sub-character classes**  SIMILAR TO and REGEXP [:upper:] and [:lower:] sub-character classes were case-insensitive on a case-insensitive database. This has been changed so that [:upper:] only matches uppercase characters and [:lower:] only matches lowercase characters, regardless of the database case sensitivity.

○ **Treatment of caret (^), underscore (_), and percent sign (%) as metacharacters**  The following table explains the previous and new treatment of these characters as metacharacters:

<table>
<thead>
<tr>
<th>Character</th>
<th>Previous behavior</th>
<th>New behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>_ (underscore)</td>
<td>For SIMILAR TO, REGEXP, and REGEXP_SUBSTR, an underscore was treated as a metacharacter: it matched any single character.</td>
<td>For SIMILAR TO, an underscore is treated as a metacharacter: it matches any single character. For REGEXP and REGEXP_SUBSTR, an underscore is not treated as a metacharacter. Instead, REGEXP, and REGEXP_SUBSTR use a period (.) to match any single character.</td>
</tr>
<tr>
<td>%</td>
<td>For SIMILAR TO, REGEXP, and REGEXP_SUBSTR, a percent sign was treated as a metacharacter: it matched any number of any characters.</td>
<td>For SIMILAR TO, a percent sign is treated as a metacharacter: it matches any number of any characters. For REGEXP and REGEXP_SUBSTR, a percent sign is not treated as a metacharacter. Instead, REGEXP, and REGEXP_SUBSTR use dot-asterisk (.*) to match any number of any characters.</td>
</tr>
</tbody>
</table>
For SIMILAR TO, REGEXP, and REGEXP_SUBSTR, a caret inside a character class was treated as a negation or subtraction character for anything to the right of it: it was interpreted as NOT matches.

For SIMILAR TO, a caret is treated as a negation or subtraction character for characters to the right of it. For example, SIMILAR TO [a-d^c] matches a, b, d, but not c.

For REGEXP and REGEXP_SUBSTR, the caret is only treated as a meta-character if it is in the first position inside a character class: it is interpreted as a negation of the character class. For example, REGEXP [^abc] matches any single character that is not a, b, or c, while REGEXP [a-d^c] matches a, b, c, d, and ^.

<table>
<thead>
<tr>
<th>Character</th>
<th>Previous behavior</th>
<th>New behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>^</td>
<td>For SIMILAR TO, REGEXP, and REGEXP_SUBSTR, a caret inside a character class was treated as a negation or subtraction character for anything to the right of it: it was interpreted as NOT matches.</td>
<td>For SIMILAR TO, a caret is treated as a negation or subtraction character for characters to the right of it. For example, SIMILAR TO [a-d^c] matches a, b, d, but not c.</td>
</tr>
</tbody>
</table>

- **Upgrade utility (dbupgrad) behavior change** Attempting to use the Upgrade utility to upgrade a database that is participating in database mirroring now results in an error. See “Upgrade utility (dbupgrad)” [SQL Anywhere Server - Database Administration].

- **Mac OS X no longer requires dbmodenv** In previous releases, to use the graphical administration tools on Mac OS X, the user’s $HOME/.MacOSX/environment.plist file had to have the SQL Anywhere binary and library location added to PATH and DYLD_LIBRARY_PATH. You could do this using the dbmodenv tool. SQL Anywhere no longer depends on the settings in $HOME/.MacOSX/environment.plist, and you no longer need to run dbmodenv or log out and log in again after installing SQL Anywhere.

- **Default changed for NULL values returned from the dbisqlc OUTPUT statement** In previous releases, if you used the OUTPUT statement from dbisqlc, the statement returned the value (NULL) for NULL values. Now, the statement returns an empty string by default for NULL values. You can change the way NULL values are exported by setting the output_nulls option. See “output_nulls option [Interactive SQL]” [SQL Anywhere Server - Database Administration].

- **Endian support** After upgrading, pre-11.0.1 text indexes created on a big-endian computer need to be truncated and refreshed (for MANUAL REFRESH and AUTO REFRESH text indexes) or recreated (for IMMEDIATE REFRESH indexes).

- **Administration tools on Mac OS X** On Mac OS X, the SQL Anywhere administration tools now use the 64-bit JDK 1.6. The administration tools only run on Intel-based MacIntosh computers with 64-bit processors supported by the Apple JDK 1.6 (Mac OS X 10.5.2 or later). If you are deploying
the administration tools for Mac OS X, the native libraries are located in $SQLANY11/System/lib64. See “Administration tool deployment on Linux, Solaris, and Mac OS X” [SQL Anywhere Server - Programming].

- **New default size for chunked transfer-coding for HTTP clients**  
  Previously, if an HTTP client sent data greater than 2048 bytes, chunked transfer-coding was attempted by default (or if the user specifies CREATE PROCEDURE...SET 'HTTP(CH=auto)'. The default size has been changed from 2048 to 8196 bytes. Also, a new status, 411 **Length Required**, has been added to the criteria for re-issuing the request without using chunked transfer-coding. See “CREATE PROCEDURE statement [Web service]” [SQL Anywhere Server - SQL Reference].

- **ansi_substring option support**  
  The ansi_substring option was deprecated in version 11.0.0, but is now supported for version 11.0.1. See “ansi_substring option” [SQL Anywhere Server - Database Administration].

### Deprecated and discontinued features

- **COMMENT ON EXTERNAL ENVIRONMENT OBJECT object-name**  
  The syntax has been changed to COMMENT ON EXTERNAL OBJECT object-name. Currently, the old syntax is still accepted but may not be supported in a future release. See “COMMENT statement” [SQL Anywhere Server - SQL Reference].

### MobiLink new features

Following is a list of additions to MobiLink introduced in version 11.0.1.

- **Schema caching of remote database schema**  
  The new schema caching feature reduces overhead for smaller synchronizations. Remote schemas are cached by the MobiLink server on the first synchronization. On subsequent synchronizations, remote schema information is only sent to the MobiLink server if it does not already have the schema cached.

- **-vi option for mlsrv11**  
  Display the column values of each row uploaded. See “-v mlsrv16 option” [MobiLink - Server Administration].

- **-vq option for mlsrv11**  
  Display the column values of each row downloaded. See “-v mlsrv16 option” [MobiLink - Server Administration].

- **-vm option for mlsrv11**  
  Prints the duration of each synchronization and the duration of each synchronization phase to the log whenever a synchronization completes. See “-v mlsrv16 option” [MobiLink - Server Administration].

- **-ppv option for mlsrv11**  
  Causes MobiLink to print new periodic monitoring values according to the period specified. See “-ppv mlsrv16 option” [MobiLink - Server Administration].

- **ml_ignore prefix**  
  The MobiLink server recognizes SQL scripts prefixed with --{ml_ignore} as intentionally ignored scripts. See “Ignored scripts” [MobiLink - Server Administration].

- **-sv option for dblsn**  
  Specifies the script version used by the MobiLink listener to authenticate against a database. See “-sv dblsn option” [MobiLink - Server-Initiated Synchronization].
● **Support for Oracle VArray**  The iAnywhere Solutions 11 - Oracle ODBC driver now supports the use of Oracle VArray in stored procedures. See “Oracle VARRAY” [MobiLink - Server Administration].

● **Light weight polling listener keywords variables**  The poll_connect, poll_notifier, poll_key, and poll_every listener keywords have been added to support light weight polling.

● **Light weight polling action variables**  $poll_connect, $poll_notifier, $poll_key, and $poll_every action variables have been added to support light weight polling.

● **Client authentication using Common Access Cards**  MobiLink clients now support authentication using client identities from Common Access Cards (CACs). See “identity_name” [MobiLink - Client Administration].

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>This feature is part of the CAC Authentication Add-on and requires a separate license. See “Separately licensed components” [SQL Anywhere 16 - Introduction].</td>
</tr>
</tbody>
</table>

● **Support for Microsoft SQL Server 2008**  The MobiLink synchronization server now supports consolidated databases running on Microsoft SQL Server 2008. For information about mapping the new Microsoft DATE, TIME, and DATETIME2 data types, see “Microsoft SQL Server data mapping” [MobiLink - Server Administration].

● **New method for .NET DownloadTableData interface**  getLastDownloadTime method to return last download time for a table. See “DownloadTableData.GetLastDownloadTime method [MobiLink server .NET]” [MobiLink - Server Administration].

● **Log verbosity for targeted MobiLink users and remote IDs**  You can now set different log verbosity for a targeted MobiLink user or remote ID. See “Log verbosity for targeted MobiLink users and remote IDs” [MobiLink - Server Administration].

● **Support for MySQL in Model mode**  The MobiLink plug-in now supports MySQL consolidated databases.

● **-c option for mimon**  The -c option has been added for the mimon command for the MobiLink Monitor. The -c option closes the MobiLink Monitor at the end of the connection and saves the session to the specified database.

### MobiLink behavior changes and deprecated features

Following is a list of changes to MobiLink introduced in version 11.0.1.

● **-sm option for mlsrv11**  The -sm option for mlsrv11 has been improved to provide similar functionality to the -nc option, when used with non-persistent HTTP/HTTPS. See “-sm mlsrv16 option” [MobiLink - Server Administration] and “-nc mlsrv16 option” [MobiLink - Server Administration].
Microsoft SQL Server data types  As of SQL Server 2005, TEXT, NTEXT, and IMAGE types are deprecated. Use VARCHAR(max), NVARCHAR(max) and VARBINARY(max) instead. See “Microsoft SQL Server data mapping” [MobiLink - Server Administration].

UltraLite new features

Following is a list of additions to UltraLite introduced in version 11.0.1.

- **Mirror file**  UltraLite provides basic database file mirroring to improve fault tolerance on potentially unreliable storage systems. This is accomplished using the mirror file. All database writes are issued to the mirror file at the same time as they are to the main database file. See “UltraLite MIRROR_FILE connection parameter” [UltraLite - Database Management and Reference].

- **Unset ml_remote_id**  You can now unset ml_remote_id using SET OPTION. See “SET OPTION statement [UltraLite]” [UltraLite - Database Management and Reference] and “ML_GET_SERVER_NOTIFICATION function [System]” [UltraLite - Database Management and Reference].

- **ML_GET_SERVER_NOTIFICATION**  This function allows UltraLite users to use light weight polling to query a notifier on a MobiLink server for server-initiated synchronization requests. See “ML_GET_SERVER_NOTIFICATION function [System]” [UltraLite - Database Management and Reference].

- **SYNC_PROFILE_OPTION_VALUE function**  This function returns the value of specified options within a sync profile. See “SYNC_PROFILE_OPTION_VALUE function [System]” [UltraLite - Database Management and Reference].

- **StreamErrorParameters property for the UltraLite.NET API**  StreamErrorParameters has been added to the SyncResult class. This member contains a comma separated list of error parameters for the stream error code reported in StreamErrorCode. See “ULSyncResult.StreamErrorParameters property [UltraLite.NET]” [UltraLite - .NET Programming].

- **getStreamErrorParameters method for the UltraLite for M-Business API**  This method returns a comma separated list of error parameters reported by the synchronization stream processing.

UltraLite behavior changes and deprecated features

Following is a list of changes to UltraLite introduced in version 11.0.1.

- **Palm Suspend functionality**  Has been deprecated.

UltraLiteJ new features

Following is a list of additions to UltraLiteJ introduced in version 11.0.1.
Additional SQL support  UltraLiteJ provides additional SQL support as follows:

○ ALTER TABLE
○ CREATE TABLE
○ CREATE INDEX
○ DROP INDEX
○ DROP TABLE
○ TRUNCATE TABLE

In addition, the following restrictions have been removed:

○ HAVING is supported.

○ DISTINCT within aggregate functions is supported.

○ CURRENT TIME, CURRENT TIMESTAMP and CURRENT DATE are supported.

See “UltraLite SQL statements” [*UltraLite - Database Management and Reference*].

New DatabaseInfo methods  Two new methods, `getPageReads()` and `getPageWrites()`, have been added to the `DatabaseInfo` interface. These methods return the number of page reads and writes at the time a `DatabaseInfo` object was created. See “DatabaseInfo.getPageReads method [UltraLiteJ]” [*UltraLite - Java Programming*] and “DatabaseInfo.getPageWrites method [UltraLiteJ]” [*UltraLite - Java Programming*].

Updated UltraLiteJ Database Transfer utility  The UltraLiteJ Database Transfer utility now provides the ability to delete a database, display database information, or view or email the database transfer log, as well as transfer a database from the client. See “UltraLite Java Edition Database Transfer utility” [*UltraLite - Database Management and Reference*].

UltraLiteJ behavior changes and deprecated features

Following is a list of changes to UltraLiteJ introduced in version 11.0.1.

○ @@identity global variable  The @@identity global variable is not supported by UltraLiteJ.

Administration tool new features

Following is a list of additions to Sybase Central and Interactive SQL introduced in version 11.0.1.
Sybase Central new features

Following is a list of additions to Sybase Central plug-ins introduced in version 11.0.1.

All plug-ins

- **Create Service Wizard enhancements**  You can now create services for the MobiLink Relay Server (rshost), relay server outbound enabler (RSOE), Volume Shadow Copy Service (dbvss11), MobiLink Listener utility (dblsn), and the Broadcast Repeater utility (dbns11) using the **Create Service Wizard**. See “How to run the database server as a service or daemon” [SQL Anywhere Server - Database Administration].

SQL Anywhere plug-in

- **View the status of events in Sybase Central**  Now you can see the current running state of events in Sybase Central. In the **Events** folder there is a new **Running** column that shows the current running state of an event. The contents of the folder are updated whenever the folder is selected in the left-pane. In addition there is a **Running** property in the **Event Properties** window. The running value shows **Yes** if the event is running, **No** if the event is not running, or **Unknown** if the event exists in a SQL Anywhere 9.0.2 or earlier database.

- **View the status of maintenance plans in Sybase Central**  Now you see the current running state of maintenance plans in Sybase Central. In the **Maintenance Plans** folder there is a new **Running** column that shows the current running state of a plan. The contents of the folder are updated whenever the folder is selected in the left-pane. The running value shows **Yes** if the maintenance plan is running, **No** if the plan is not running, or **Unknown** if the plan exists in a SQL Anywhere 9.0.2 or earlier database.

- **Maintenance plan enhancements**  The **Create Maintenance Plan Wizard** now lets you send a test email when you configure the settings to email a maintenance plan report. See “Maintenance plans” [SQL Anywhere Server - Database Administration].

- **Create Procedure Wizard and Create Function Wizard enhancements**  Now in the **Create Function Wizard** and the **Create Procedure Wizard**, you can choose one of the following SQL dialects or languages to write the procedure or function in: Watcom-SQL, Transact-SQL, External C/C++, or External Environment. If you choose External Environment, you select one of the following languages: C_ESQL32, C_ESQL64,C_ODBC32, C_ODBC64, CLR, JAVA, PERL, or PHP. Previously, only Watcom-SQL or Transact-SQL were available choices. Choosing C/C++ or Java generates a code skeleton for the function or procedure with the EXTERNAL NAME clause. See “CREATE FUNCTION statement [External call]” [SQL Anywhere Server - SQL Reference] and “CREATE PROCEDURE statement [External call]” [SQL Anywhere Server - SQL Reference].

MobiLink plug-in

- **MobiLink model enhancement for Oracle databases**  Now, when you use the **Create Synchronization Model Wizard** to create a synchronization model that uses an Oracle consolidated database, you can choose to load the entire schema, or to select a subset of owners whose tables you need for synchronization. Choosing a subset of owners can reduce the schema loading time.
MobiLink Model mode support for MySQL  MobiLink Model mode now supports the use of MySQL consolidated databases.

Administration tool behavior changes and deprecated features

Following is a list of changes to Sybase Central and Interactive SQL introduced in version 11.0.1.

Sybase Central behavior changes and deprecated features

Following is a list of changes to Sybase Central introduced in version 11.0.1.

SQL Anywhere plug-in

- Read-only database enhancement  Now you receive a warning when you connect to a read-only database. You can choose to suppress this warning by clicking Tools » SQL Anywhere 11 » Preferences.

Interactive SQL behavior changes and deprecated features

Following is a list of changes to Interactive SQL introduced in version 11.0.1.

- Configure the automatic release of database locks  Now, you can configure Interactive SQL to attempt to release the database schema locks it creates when it displays your result set. To do so, in Interactive SQL, click Tools » Options » SQL Anywhere, and click Automatically Release Database Locks.

  When this option is selected, after you execute a statement that returns a result set, Interactive SQL checks if your connection has any uncommitted changes in the database. If none exist, then Interactive SQL releases your schema locks; otherwise, Interactive SQL does not release your schema locks. That is, Interactive SQL does not release your schema locks if you have any uncommitted changes to the database.

Product-wide new features

Following is a list of product-wide additions introduced in version 11.0.1.

- New online documentation forum, DocCommentXchange (DCX)  A new online forum called DocCommentXchange has been created. DocCommentXchange is a community for accessing and discussing SQL Anywhere documentation.

  Use DocCommentXchange to:

  ○ View documentation
○ Check for clarifications users have made to sections of documentation

○ Provide suggestions and corrections to improve documentation for all users in future releases


- **SQL Anywhere Monitor**  The SQL Anywhere Monitor is a browser-based administration tool that provides you with information about the health and availability of SQL Anywhere databases and MobiLink servers.

For MobiLink users, this feature does not replace or overlap the functionality in the existing MobiLink Monitor.

See “SQL Anywhere Monitor” [SQL Anywhere Server - Database Administration].

## Documentation enhancements

- **SQL Anywhere backup and recovery documentation enhancements**  The documentation for using the backup and recovery tools included with SQL Anywhere has been re-written for this release. See “Backup and data recovery” [SQL Anywhere Server - Database Administration].

- **Full text search documentation enhancements**  The documentation for the full text feature has been reorganized and now contains more examples and tutorials. See “Full text search” [SQL Anywhere Server - SQL Usage].

- **MobiLink Server-Initiated Synchronization book**  The MobiLink server-initiated synchronization book has undergone improvements since version 11.0.0. For example, the table of contents organization has been improved, server-initiated synchronization is explained in more detail, and improvements have been made to the format of the reference material. See “MobiLink - Server-Initiated Synchronization”.

- **SQL Remote book**  The SQL Remote book has been reorganized and re-written to enhance navigation and usability. See “SQL Remote”. 
What's new in version 11.0.0


Note
As with all forward-looking statements, the lists of deprecated features are not guaranteed to be complete and are subject to change.

SQL Anywhere
- “SQL Anywhere new features” on page 206
- “SQL Anywhere behavior changes” on page 232
- “SQL Anywhere deprecated and discontinued features” on page 240

MobiLink
- “MobiLink new features” on page 244
- “MobiLink behavior changes and deprecated features” on page 248

SQL Remote
- “SQL Remote new features” on page 249
- “SQL Remote behavior changes and deprecated features” on page 249

UltraLite
- “UltraLite new features” on page 249
- “UltraLite behavior changes and deprecated features” on page 254

Sybase Central and Interactive SQL
- “Sybase Central and Interactive SQL new features” on page 256
- “Sybase Central and Interactive SQL behavior changes and deprecated features” on page 259

Documentation enhancements
- “Documentation enhancements” on page 264

Product-wide features
- “Product-wide new features” on page 265
- “Product-wide behavior changes” on page 265

SQL Anywhere

The following sections describe the new features, behavior changes, and deprecated features in SQL Anywhere for version 11.0.0.
SQL Anywhere new features

Following is a list of additions to SQL Anywhere databases and database servers introduced in version 11.0.0.

Main features

Following is a list of the main features introduced in SQL Anywhere version 11.0.0.

- **Support for merging tables** SQL Anywhere now allows you to merge tables, views, and system procedure results into a table or view. See “MERGE statement” [SQL Anywhere Server - SQL Reference].

- **Support for login policies** SQL Anywhere now supports login policies. A login policy is a set of options that define rules to be applied when a database connection is established for a user. SQL Anywhere supplies a root policy used to store default values for all login policies. Separate login policies can be created to store overrides to the root login policy options. Each user is assigned a login policy and does not inherit any of the login policies through group memberships. The assignment of a policy to a specific user can be modified as necessary. See “Login policies” [SQL Anywhere Server - Database Administration].

  A database upgrade is required to take advantage of this feature. See “SQL Anywhere Server upgrades” on page 272.

- **Support for full text search** SQL Anywhere now supports full text search. Full text search can quickly find all instances of a term (word) in a database. Full text search differs from searching using predicates such as LIKE, REGEXP, and SIMILAR TO because it is term-based and because it uses a text index instead of scanning table rows. See “Full text search” [SQL Anywhere Server - SQL Usage].

  You must upgrade your database to make use of the full text search feature. See “SQL Anywhere Server upgrades” on page 272.

- **Support for regular expressions** SQL Anywhere now provides support for regular expressions using two new search conditions, REGEXP and SIMILAR TO. See “REGEXP search condition” [SQL Anywhere Server - SQL Reference] and “SIMILAR TO search condition” [SQL Anywhere Server - SQL Reference].

  See also, “Regular expressions overview” [SQL Anywhere Server - SQL Reference] and “LIKE, REGEXP, and SIMILAR TO search conditions” [SQL Anywhere Server - SQL Reference].

- **Database option settings are now recorded in the transaction log** The database option settings in effect during a LOAD operation are now recorded in the transaction log. This ensures that there are no inconsistencies in the data should options, such as date_order and nearest_century, change between the original LOAD operation and the final LOAD operation resulting from applying the transaction log during recovery.

- **Enhancements to the optimizer's use of indexes** Several enhancements have been made to the indexing capabilities of SQL Anywhere. You must upgrade your database to make use of these new features. See “SQL Anywhere Server upgrades” on page 272.
Support for multiple indexes scan  The optimizer has been enhanced to consider multiple
indexes (up to four) to retrieve data from a base table based on multiple predicates on that table.
Previously, you could only specify one index as an index hint for a query. A new index hint in the
WITH clause of the SELECT statement allows you to specify that a multiple index scan can be
used. See “FROM clause” [SQL Anywhere Server - SQL Reference].

A new table access algorithm, Multiple Index Scan, has been added.

Support for index-only retrieval  The optimizer has been enhanced to support index-only
retrievals. With index-only retrieval, the query is satisfied using data from the indexes, without
having to access corresponding rows in the tables. The optimizer always performs an index-only
retrieval when possible. An INDEX ONLY { ON | OFF } hint can be used to control whether
index-only retrieval is performed. See “FROM clause” [SQL Anywhere Server - SQL Reference].

Enhancements to loading and unloading data  The following enhancements have been made to
loading and unloading data:

Load data from, and unload data to, files on a client computer  SQL statements and
functions are used to read and write data residing on the database server. New features have been
implemented to extend this capability to files that reside on the client computer, without the need
to copy client files onto the database server. The transfer of data is accomplished efficiently, while
providing security and access control for data on the client computer.

The actual reading of files on the client computer is done transparently by the client libraries,
which means that existing client applications can start benefitting immediately from the new
feature using the new SQL language support.

To benefit from these new capabilities, both the client and the database server must be SQL
Anywhere version 11.0.0, and the client must use the Command Sequence communication
protocol (CmdSeq).

See “Access to data on client computers” [SQL Anywhere Server - SQL Usage].

Unload data into a variable  The UNLOAD statement has been enhanced to include an INTO
VARIABLE clause to allow you to unload data into a variable. See “UNLOAD statement” [SQL
Anywhere Server - SQL Reference].

Load data from a column in another table  The LOAD TABLE statement has been
enhanced to include a USING COLUMN clause to allow you to load data from a column in
another table. See “LOAD TABLE statement” [SQL Anywhere Server - SQL Reference] and “Data
import with the LOAD TABLE statement” [SQL Anywhere Server - SQL Usage].

A database upgrade is required to take advantage of this new feature. See “SQL Anywhere Server
upgrades” on page 272.

Load data from a value (BLOB)  The LOAD TABLE statement has been enhanced to include
a USING VALUE clause to allow you to load data from a value expression, such as the results of
a function or a system procedure. See “LOAD TABLE statement” [SQL Anywhere Server - SQL
Reference], and “Data import with the LOAD TABLE statement” [SQL Anywhere Server - SQL
Usage].
A database upgrade is required to take advantage of this new feature. See “SQL Anywhere Server upgrades” on page 272.

- **LOAD TABLE statement recovery and mirroring enhancements**  
  Previously, in a mirrored database configuration, loading data from a file using the LOAD TABLE statement was not supported because only the LOAD TABLE statement was recorded in the transaction log, not the data being loaded. Additionally, when recovering a database, data loaded using a LOAD TABLE statement was not recoverable unless the original load file was present during recovery.

  The LOAD TABLE statement has been enhanced to include three new logging option clauses: WITH CONTENT LOGGING, WITH ROW LOGGING, and WITH FILE NAME LOGGING. These clauses allow you to control whether to record the loaded data in the transaction log. In a database mirroring system, that data can be used to load the mirror database. Additionally, during recovery, the load file no longer needs to be present. See “LOAD TABLE statement” [SQL Anywhere Server - SQL Reference].

  A database upgrade is required to take advantage of this feature. See “SQL Anywhere Server upgrades” on page 272.

- **Enhancements to materialized views**  
  Support for materialized views has been enhanced as follows:

  - **Support for immediate materialized views**  
    You can now configure materialized views to be refreshed immediately when data changes in the underlying tables impact data in the materialized view. Views with this refresh type are called immediate views; views that are not refreshed immediately are now referred to as manual views. Materialized views created before this release are considered manual views, and are the default when creating a new materialized view.

  For more information about manual and immediate views, see “Materialized views” [SQL Anywhere Server - SQL Usage] and “Whether to set refresh type to manual or immediate” [SQL Anywhere Server - SQL Usage].

  You must upgrade your database to use the system procedures that support this feature. See “SQL Anywhere Server upgrades” on page 272.

  - **Ability to refresh multiple materialized views at once**  
    Previously, you needed to refresh materialized views one at a time. Changes to underlying data between each refresh operation could introduce inconsistencies between the materialized views. Now, to refresh multiple materialized views using the same data, you can specify a list of materialized views for the REFRESH MATERIALIZED VIEW statement. See “REFRESH MATERIALIZED VIEW statement” [SQL Anywhere Server - SQL Reference] and “Manually refreshing a materialized view” [SQL Anywhere Server - SQL Usage].

  - **New WITH SHARE MODE clause for the REFRESH MATERIALIZED VIEW statement**  
    A new clause, WITH SHARE MODE, has been added to the REFRESH MATERIALIZED VIEW statement. This mode gives read access on underlying tables to other transactions while the refresh operation takes place. When this clause is specified, shared table locks are obtained on all underlying base tables before the refresh operation is performed. The default mode is now WITH SHARE MODE, unless the materialized view is defined as IMMEDIATE REFRESH, or snapshot
isolation is enabled for the database. For more information about the default refresh behavior, see “REFRESH MATERIALIZED VIEW statement” [SQL Anywhere Server - SQL Reference].

- **Support for querying the contents of a file or BLOB string** Using the new OPENSTRING subclause of the FROM clause, you can now query data from a file or a BLOB string. The OPENSTRING clause allows you to specify the object to be queried, as well as the schema and other parsing information for the data. See “FROM clause” [SQL Anywhere Server - SQL Reference].

  A new plan item, OpenString, appears in the execution plan when the OPENSTRING operation is performed.

- **Improved support for compressed indexes** Because of the work to improve support for compressed indexes, when you rebuild a database by unloading and reloading it, the rebuilt database may be smaller than the original database. This decrease in database size does not indicate a problem or a loss of data.

- **Read-only access to databases running on a mirror server** If you are using database mirroring, you can now connect to the database running on the mirror server. This enables you to offload potentially resource-heavy reporting operations to the mirror server, while leaving the primary server available. You can connect to a mirror database by providing a database server name with the -sm server option that can be used to access the read-only mirror database. See “-sm database option (deprecated)” [SQL Anywhere Server - Database Administration].

### Database connections

Following is a list of enhancements made to database connections in SQL Anywhere version 11.0.0.

- **AppInfo connection parameter enhancements** The AppInfo connection parameter now supports the OSUSER key. This key returns the operating system user name associated with the client process. Linux and Solaris now support the EXE key. See “AppInfo (APP) connection parameter” [SQL Anywhere Server - Database Administration].

- **Elevate connection parameter** The Elevate connection parameter elevates automatically started SQL Anywhere database servers on Windows Vista. See “Elevate connection parameter” [SQL Anywhere Server - Database Administration].

- **NewPassword connection parameter [NEWPWD]** The NewPassword connection parameter allows users to change passwords, even if they have expired, without DBA intervention. See “NewPassword (NEWPWD) connection parameter” [SQL Anywhere Server - Database Administration].

- **Prefetch enhancements** The default values for the PrefetchBuffer (PBUF) connection parameter have changed. On Windows Mobile, the default value is now 64 KB, and on all other platforms the default value is now 512 KB. This connection parameter accepts values between 64 KB and 8 MB. See “PrefetchBuffer (PBUF) connection parameter” [SQL Anywhere Server - Database Administration].

In previous versions, the maximum number of prefetched rows was based on the maximum amount of data that could be prefetched. Now, the maximum number of prefetched rows takes into account the
actual amount of prefetched data, as well as the data limit specified by the PrefetchBuffer connection parameter. This can result in significant performance gains when the amount of data in a column is significantly less than both the host variable length and describe length.

Prefetch also dynamically increases the number of prefetch rows in situations that are likely to result in improved performance. See “Prefetches” [SQL Anywhere Server - Programming].

Backup and recovery
Following is a list of backup and recovery enhancements introduced in SQL Anywhere version 11.0.0.

- **Support for the Microsoft Volume Shadow Copy Service (VSS)** SQL Anywhere is compatible with Microsoft Volume Shadow Copy Service (VSS). To use VSS, you must rebuild all existing databases. See “SQL Anywhere Volume Shadow Copy Service (VSS)” [SQL Anywhere Server - Database Administration].

Security
Following is a list of security enhancements introduced in SQL Anywhere version 11.0.0.

- **ISYSUSER and ISYSEXTERNLOGIN system tables are now encrypted when table encryption is enabled** Previously, when encrypting a database, or when creating a database with table encryption enabled, the ISYSCOLSTAT system table was automatically encrypted. Now, the ISYSUSER and ISYSEXTERNLOGIN system tables are also encrypted, to provide additional security.

- **Auditing enhancements** Now, auditing can be controlled through Sybase Central. From the Database Properties window, you can enable auditing, disable auditing, and specify which information they want to audit. Auditing information can be viewed in Sybase Central on the Auditing tab in the right pane. See “Configuring auditing (Sybase Central)” [SQL Anywhere Server - Database Administration] and “Retrieving auditing information (Sybase Central)” [SQL Anywhere Server - Database Administration].

  When auditing is enabled, errors for failed connections are now logged, indicating the reason for the failure.

- **256-bit AES encryption now supported** SQL Anywhere now supports 256-bit AES encryption for databases, tables, files, and data. This enhancement impacts several areas, as noted below:

  - **Database and table encryption** You can now specify AES256 and AES256_FIPS for the ENCRYPTION clause of the CREATE DATABASE statement. See “CREATE DATABASE statement” [SQL Anywhere Server - SQL Reference].

    You can also specify AES256 and AES256_FIPS for the -ea option of the Initialization utility (dbinit) and Unload utility (dbunload). See “Initialization utility (dbinit)” [SQL Anywhere Server - Database Administration] and “Unload utility (dbunload)” [SQL Anywhere Server - Database Administration].
○ **FIPS-certified algorithms** You can now use 256-bit FIPS-certified AES algorithms. See “-fips database server option” [SQL Anywhere Server - Database Administration].

○ **Encrypting and decrypting data** When encrypting data using the ENCRYPT and DECRYPT functions, you can now specify AES256 and AES256_FIPS. See “ENCRYPT function [String]” [SQL Anywhere Server - SQL Reference] and “DECRYPT function [String]” [SQL Anywhere Server - SQL Reference].

○ **Creating encrypted copies of databases, transaction logs, and dbspaces** When creating an encrypted copy of an encrypted or unencrypted database, transaction log, or dbspace using the CREATE ENCRYPTED FILE statement, you can now specify a 256-bit AES algorithm (AES256 or AES256_FIPS). See “CREATE ENCRYPTED FILE statement” [SQL Anywhere Server - SQL Reference].

○ **DBTools support for 256-bit AES encryption** The a_create_db and an_unload_db structures have been extended to support AES256 and AES256_FIPS as values for the encryption_algorithm member. See “a_create_db structure [database tools]” [SQL Anywhere Server - Programming] and “an_unload_db structure [database tools]” [SQL Anywhere Server - Programming].

See also:

○ “Database encryption and decryption” [SQL Anywhere Server - Database Administration]
○ “List of database properties” [SQL Anywhere Server - Database Administration]

● **Password encryption supported for jConnect and Open Client** Password encryption is now supported for jConnect and Open Client connections. See:

  ○ “The jConnect JDBC driver” [SQL Anywhere Server - Programming]
  ○ “JDBC client deployment” [SQL Anywhere Server - Programming]
  ○ “Known Open Client limitations of SQL Anywhere” [SQL Anywhere Server - Programming]

**Security changes**

You must upgrade your database to make use of these changes. See “SQL Anywhere Server upgrades” on page 272.

● **Support for inheritance** SQL Anywhere now supports the inheritance of the PROFILE, READCLIENTFILE, READFILE, and WRITECLIENTFILE authorities.

● **New authorities:**

  ○ PROFILE
  ○ READFILE
  ○ READCLIENTFILE
  ○ WRITECLIENTFILE

● **CREATE privilege supported for dbspaces** SQL Anywhere now supports the CREATE ON permission for creating database object on the specified dbspace. The CREATE privilege can be assigned to the user, or inherited through group membership.
# Database utilities

Following is a list of enhancements made to database utilities in SQL Anywhere version 11.0.0.

- **Configuration file enhancement**  You can now use an ampersand (&) in configuration files to indicate that the previous token is continued on the next line. See “Configuration files” [SQL Anywhere Server - Database Administration].

- **Unload utility (dbunload) enhancements**  The following enhancements have been made to dbunload:
  
  - A new option, -cp, has been added to allow you to cause dbunload to compress the data output file.
  
  - Previously, if specified the -ek, -ep, or -ea encryption options, without also specifying the -an or -ar reload options, an error was returned. Now, however, dbunload accepts the encryption options, and applies them to the output file it creates.
  
  - The -g option now refreshes text indexes defined as MANUAL REFRESH. See “Text index concepts and reference” [SQL Anywhere Server - SQL Usage].
  
  - By default, text indexes defined as MANUAL REFRESH are not initialized as part of a reload. If you want to initialize these text indexes, you can specify the dbunload -g option.
  
  - The -no option lets you unload object definitions in alphabetical order, grouped by object type. This can be useful for comparing the reload.sql files for databases.

  See “Unload utility (dbunload)” [SQL Anywhere Server - Database Administration].

- **Validation utility (dbvalid) enhancements**  Previously, the dbvalid utility validated all tables and materialized views by default. Dbvalid now also executes a VALIDATE DATABASE statement.

  When a database is started automatically by running the Validation utility, it is started in read-only mode. This prevents you from changing the database if it is being validated as part of a backup and recovery plan.

  See “Validation utility (dbvalid)” [SQL Anywhere Server - Database Administration].

- **Maintaining the generated .sql file if the Log Translation utility (dbtran) detects corruption**  The Log Translation utility -k option lets you specify that you do not want a partial .sql file deleted if translating the log file fails because of corruption in the transaction log file. See “Log Translation utility (dbtran)” [SQL Anywhere Server - Database Administration].

# Database options

Following is a list of enhancements made to database options in SQL Anywhere version 11.0.0.

- **allow_read_client_file option**  This option controls whether to allow the reading of files on a client computer. See “allow_read_client_file option” [SQL Anywhere Server - Database Administration].
- **allow_write_client_file option**  This option controls whether to allow the writing of files to a client computer. See “allow_write_client_file option” [SQL Anywhere Server - Database Administration].

- **login_procedure option**  You can now signal an expired password error message. See “login_procedure option” [SQL Anywhere Server - Database Administration].

- **max_priority option**  This option controls the maximum priority level for database connections. See “max_priority option” [SQL Anywhere Server - Database Administration].

- **priority option**  This option controls the priority level at which requests from a connection are executed. See “priority option” [SQL Anywhere Server - Database Administration].

- **query_mem_timeout option**  This option controls how long a request waits for a memory grant. See “query_mem_timeout option” [SQL Anywhere Server - Database Administration].

### Database server options

Following is a list of enhancements made to database server options in SQL Anywhere version 11.0.0.

- **-es server option**  In previous versions of SQL Anywhere, if the database server was started with the -ec option (to support transport layer security), but the list of allowed encryption protocols did not include NONE or SIMPLE, then the shared memory port was not started because it does not support transport layer security. This meant that all connections to the database server had to be made over TCP/IP using strong encryption.

  The -es server option instructs the database server to allow unencrypted connections over shared memory. See “-es database server option” [SQL Anywhere Server - Database Administration].

- **-gb server option**  The -gb server option for controlling the server process priority class is now supported on Unix, as well as Windows. See “-gb database server option” [SQL Anywhere Server - Database Administration].

- **-im server option**  You can run the database entirely in memory if the loss of all database operations is tolerable for your application. This feature is intended for situations where SQL Anywhere is intended to be used as a fast, temporary data-store, where data is inserted at a rapid rate. See “-im database server option” [SQL Anywhere Server - Database Administration].

- **Reading from and writing to files on a client computer**  The -sf server option now allows you to control the ability to read from, and write to, files on a client computer. See “-sf database server option” [SQL Anywhere Server - Database Administration].

- **-um server option**  The -um option allows you to connect to the DBLauncher.app instance, if it is running, and display database server messages in a new window within DBLauncher.app. This option only applies to Mac OS X. See “-um database server option” [SQL Anywhere Server - Database Administration].

- **Windows Performance Monitor options**  The following server options have been added to further configure the Windows Performance Monitor:
Properties and Performance Monitor statistics

Following is a list of enhancements made to properties and Performance Monitor statistics in SQL Anywhere version 11.0.0.

- **New connection properties**  
  The following connection properties have been added in this release:

  - allow_read_client_file
  - allow_write_client_file
  - AuthType
  - CacheReadWorkTable
  - ClientNodeAddress
  - DiskReadWorkTable
  - DiskSyncRead
  - DiskSyncWrite
  - DiskWaitRead
  - DiskWaitWrite
  - DiskWriteHint
  - DiskWriteHintPages
  - LockIndexID
  - LockRowID
  - max_priority
  - OSUser
  - priority
  - query_mem_timeout
  - QueryMemActiveCurr
  - QueryMemExtraAvail
  - QueryMemGrantFailed
  - QueryMemGrantGranted
  - QueryMemGrantWaiting
  - QueryMemGrantRequested
  - QueryMemWaited
  - ServerNodeAddress
  - ReadHint
  - ReadHintScatter

  For descriptions of these properties, see “List of connection properties” [SQL Anywhere Server - Database Administration].
The following database server properties have been added in this release:

- DiskRetryRead
- DiskRetryReadScatter
- DiskRetryWrite
- EventTypeDesc
- EventTypeName
- HttpAddresses
- HttpsAddresses
- HttpNumActiveReq
- HttpNumConnections
- HttpNumSessions
- HttpsNumActiveReq
- HttpsNumConnections
- MaxEventType
- MaxRemoteCapability
- MessageCategoryLimit
- OptionWatchAction
- OptionWatchList
- QueryMemActiveCurr
- QueryMemActiveEst
- QueryMemActiveMax
- QueryMemExtraAvail
- QueryMemGrantBase
- QueryMemGrantBaseMI
- QueryMemGrantExtra
- QueryMemGrantFailed
- QueryMemGrantGranted
- QueryMemGrantWaiting
- QueryMemGrantRequested
- QueryMemPages
- QueryMemPercentOfCache
- QueryMemWaited
- ReadHintScatterLimit
- RemoteCapability
- StreamsUsed
- TcpIpAddresses
- WebClientLogFile
- WebClientLogging

For descriptions of these properties, see “List of database server properties” [SQL Anywhere Server - Database Administration].
● **New database properties**  The following database properties have been added in this release:

- AlternateMirrorServerName
- CacheReadWorkTable
- DiskReadWorkTable
- DiskRetryReadScatter
- DiskSyncRead
- DiskSyncWrite
- DiskWaitRead
- DiskWaitWrite
- DiskWriteHint
- DiskWriteHintPages
- HasEndianSwapFix
- MirrorMode
- ReadHint
- ReadHintScatter

For descriptions of these properties, see “List of database properties” [SQL Anywhere Server - Database Administration].

● **New Performance Monitor statistics**  The following Performance Monitor statistics have been added in this release:

- Cache Reads: Work Table
- Disk Reads: Work Table

---

### System procedures and functions

Following is a list of system procedure and function enhancements added in SQL Anywhere version 11.0.0.

● **sa_get_dtt_groupreads system procedure**  The new sa_get_dtt_groupreads system procedure allows you to estimate the cost of issuing group reads on the database server. See “sa_get_dtt_groupreads system procedure” [SQL Anywhere Server - SQL Reference].

● **PROPERTY_NAME function enhancement**  Now returns the name of the property with the supplied property ID for the specified connection level. See “PROPERTY_NAME function [System]” [SQL Anywhere Server - SQL Reference].

● **READ_CLIENT_FILE function**  The new READ_CLIENT_FILE function reads data from the specified file on the client computer. See “READ_CLIENT_FILE function [String]” [SQL Anywhere Server - SQL Reference].

● **WRITE_CLIENT_FILE function**  The new WRITE_CLIENT_FILE function writes data to the specified file on the client computer. See “WRITE_CLIENT_FILE function [String]” [SQL Anywhere Server - SQL Reference].
- **REGEXP_SUBSTR function**  The new REGEXP_SUBSTR function allows you to search for a substring within a string. This new function takes a regular expression as an argument. See “REGEXP_SUBSTR function [String]” [SQL Anywhere Server - SQL Reference].

- **sa_char_terms system procedure**  The new sa_char_terms system procedure breaks a CHAR string into terms and returns every term together with its position. See “sa_char_terms system procedure” [SQL Anywhere Server - SQL Reference].

- **sa_nchar_terms system procedure**  The new sa_nchar_terms system procedure breaks an NCHAR string into terms and returns every term together with its position. See “sa_nchar_terms system procedure” [SQL Anywhere Server - SQL Reference].

- **sa_refresh_text_indexes system procedure**  The new sa_refresh_text_indexes system procedure refreshes all text indexes that are defined as MANUAL REFRESH or AUTO REFRESH. See “sa_refresh_text_indexes system procedure” [SQL Anywhere Server - SQL Reference].

- **sa_text_index_stats system procedure**  The new sa_text_index_stats system procedure returns statistical information for all text indexes in the database, including the last refresh time and size of pending changes. See “sa_text_index_stats system procedure” [SQL Anywhere Server - SQL Reference].

- **sa_text_index_vocab system procedure**  The new sa_text_index_vocab system procedure lists all terms that appear in a text index, and the total number of indexed values that each term appears in. See “sa_text_index_vocab system procedure” [SQL Anywhere Server - SQL Reference].

Two new system procedures, sa_internal_text_index_vocab and sa_internal_text_index_postings have also been added, but are only for use by the sa_text_index_vocab system procedure.

- **sa_text_index_postings system procedure**  This new system procedure is for internal use only.

- **sa_text_index_handles system procedure**  This new system procedure is for internal use only.

- **sa_get_user_status system procedure**  The new sa_get_user_status system procedure allows you to determine a user's current login status. See “sa_get_user_status system procedure” [SQL Anywhere Server - SQL Reference].

- **Running procedures and functions as invoker**  When creating a procedure or function, you can now specify whether the procedure or function run with the permissions of the user calling it (invoker), or by the owner (definer). To specify this, use the SQL SECURITY clause of the CREATE PROCEDURE or CREATE FUNCTION statement. See “CREATE FUNCTION statement” [SQL Anywhere Server - SQL Reference] and “CREATE PROCEDURE statement” [SQL Anywhere Server - SQL Reference].

This change also applies to external procedures and functions.

- **sa_disk_free_space system procedure**  The sa_disk_free_space system procedure now returns a new column, total_space, indicating the total amount of disk space available on the drive where the dbspace resides. For databases created on versions of SQL Anywhere before 11.0.0, the total_space column is not returned until the database is upgraded. See “sa_disk_free_space system procedure” [SQL Anywhere Server - SQL Reference].
• **sa_external_library_unload system procedure** A new system procedure, `sa_external_library_unload`, has been added to allow you to unload external libraries that are not in use. See “sa_external_library_unload system procedure” [SQL Anywhere Server - SQL Reference].

• **sa_index_density system procedure now returns skew** The `sa_index_density` system procedure has been enhanced to return the amount of skew present in the index. A high degree of skew can impact performance compared to a well balanced index. See “Reduce index fragmentation and skew” [SQL Anywhere Server - SQL Usage] and “sa_index_density system procedure” [SQL Anywhere Server - SQL Reference].

• **sa_materialized_view_info system procedure enhancements** Information in the Status column returned by `sa_materialized_view_info` has been split into two columns, Status and DataStatus. The Status now returns information on whether the view is enabled or disabled. The new DataStatus column returns information about whether there is data in the view, and the freshness of the data. An additional column, RefreshType, has been added to indicate whether the view is a manual view or an immediate view. See “sa_materialized_view_info system procedure” [SQL Anywhere Server - SQL Reference].

• **sa_materialized_view_can_be_immediate system procedure** Newly created materialized views are manual views by default, but can be altered to become immediate views, providing they do not violate any of the restrictions for immediate views. The new `sa_materialized_view_can_be_immediate` system procedure allows you to test whether a manual view can be changed to an immediate view. See “sa_materialized_view_can_be_immediate system procedure” [SQL Anywhere Server - SQL Reference] and “Restrictions when changing a materialized view from manual to immediate” [SQL Anywhere Server - SQL Usage].

• **sa_post_login_procedure system procedure** A new system procedure has been added to allow you to determine if a warning should be issued when a user's password is about to expire. See “sa_post_login_procedure system procedure” [SQL Anywhere Server - SQL Reference].

• **EVENT_PARAMETER function enhancement** The `EVENT_PARAMETER` function now supports abnormal as a DisconnectReason. This new reason indicates that a disconnect occurred either as a result of the client application shutting down abnormally before disconnecting from the database, or as a result of a communication failure between the client and server computers. See “EVENT_PARAMETER function [System]” [SQL Anywhere Server - SQL Reference].

• **sa_server_option system procedure enhancements** Two new properties, OptionWatchList and OptionWatchAction, have been added to the `sa_server_option` system procedure. You can use these properties to monitor when an attempt is made to change a database option setting, and specify the action to take. See “Monitoring option settings” [SQL Anywhere Server - Database Administration] and “sa_server_option system procedure” [SQL Anywhere Server - SQL Reference].

• **sa_db_properties system procedure enhancement** The `sa_db_properties` system procedure now returns valid properties that have NULL values. See “sa_db_properties system procedure” [SQL Anywhere Server - SQL Reference].

• **sa_conn_properties system procedure enhancement** The `sa_conn_properties` system procedure now returns valid properties that have NULL values. See “sa_conn_properties system procedure” [SQL Anywhere Server - SQL Reference].
SQL statements

Following is a list of SQL enhancements introduced in SQL Anywhere version 11.0.0.

- **New CALIBRATE GROUP READ clause, ALTER DATABASE statement** The new CALIBRATE GROUP READ clause of the ALTER DATABASE statement allows you to perform group read calibration on the temporary dbspace. See “ALTER DATABASE statement” [SQL Anywhere Server - SQL Reference].

- **New CHECK clause, CREATE MATERIALIZED VIEW statement** The new CHECK clause of the CREATE MATERIALIZED VIEW statement allows you to validate the statement before creating the view. See “CREATE MATERIALIZED VIEW statement” [SQL Anywhere Server - SQL Reference].

- **New RECOMPILE clause, ALTER FUNCTION statement** A new clause, RECOMPILE, has been added to the ALTER FUNCTION statement to allow you to recompile a user-defined function. See “ALTER FUNCTION statement” [SQL Anywhere Server - SQL Reference].

- **New RECOMPILE clause, ALTER PROCEDURE statement** A new clause, RECOMPILE, has been added to the ALTER PROCEDURE statement to allow you to recompile a stored procedure. See “ALTER PROCEDURE statement” [SQL Anywhere Server - SQL Reference].

- **New REFRESH clause, ALTER MATERIALIZED VIEW statement** A new clause, REFRESH, has been added to the ALTER MATERIALIZED VIEW statement to allow you to specify the refresh type for the materialized view. See “ALTER MATERIALIZED VIEW statement” [SQL Anywhere Server - SQL Reference].

- **LOAD TABLE statement enhancements in support of recovery and mirroring** The following clauses have been added to the LOAD TABLE statement in support of recovery and mirroring:
  
  - **WITH CONTENT LOGGING clause** The WITH CONTENT LOGGING clause instructs the database server to record the contents of the data source in the transaction log. The data is recorded in small chunks as the input is processed by LOAD TABLE. These chunks can be reconstituted into rows by a mirroring database, or when recovering from the transaction log. The WITH CONTENT LOGGING clause can be beneficial when it is not desirable to maintain the original data files for later recovery. See “LOAD TABLE statement” [SQL Anywhere Server - SQL Reference].

  - **WITH ROW LOGGING clause** The WITH ROW LOGGING clause instructs the database server to record, as a series of INSERT statements, all the rows being loaded. This level is ideal for databases involved in synchronization, as well as in situations where the table being loaded into contains non-deterministic values, such as computed columns, or CURRENT TIMESTAMP defaults. See “LOAD TABLE statement” [SQL Anywhere Server - SQL Reference].

  - **WITH FILE NAME LOGGING clause** The WITH FILE NAME LOGGING clause instructs the database server to record only the LOAD TABLE statement. This is the default behavior and is consistent with the logging behavior in previous versions of SQL Anywhere. See “LOAD TABLE statement” [SQL Anywhere Server - SQL Reference].
• New client file loading and unloading clauses  The enhancements have been made to the LOAD TABLE and UNLOAD TABLE statements in support of the new client file loading/unloading feature:

  ○ New USING CLIENT FILE clause for the LOAD TABLE statement  Allows you to load a table using data in a file located on the client computer. See “LOAD TABLE statement” [SQL Anywhere Server - SQL Reference].

  ○ New INTO CLIENT FILE clause for the UNLOAD TABLE statement  Allows you to specify a file on the client computer to unload data into. See “UNLOAD statement” [SQL Anywhere Server - SQL Reference].

• New login policy statements  The following statements have been added in support of the new login policy feature:

  ○ CREATE LOGIN POLICY statement. See “CREATE LOGIN POLICY statement” [SQL Anywhere Server - SQL Reference].

  ○ ALTER LOGIN POLICY statement. See “ALTER LOGIN POLICY statement” [SQL Anywhere Server - SQL Reference].

  ○ DROP LOGIN POLICY statement. See “DROP LOGIN POLICY statement” [SQL Anywhere Server - SQL Reference].

  ○ CREATE USER statement. See “CREATE USER statement” [SQL Anywhere Server - SQL Reference].

  ○ ALTER USER statement. See “ALTER USER statement” [SQL Anywhere Server - SQL Reference].

  ○ DROP USER statement. See “DROP USER statement” [SQL Anywhere Server - SQL Reference].

• New full text search statements and clauses  The following statements have been added in support of the new full text search feature:

  ○ New CONTAINS search condition  The CONTAINS search condition is used to check a specified list of columns for the existence of a specified list of terms or phrases. The CONTAINS search condition returns either TRUE or FALSE. When searching for multiple terms or phrases, you can combine them with various Boolean operators. See “CONTAINS search condition” [SQL Anywhere Server - SQL Reference].

  ○ New CONTAINS clause in the FROM clause of a SELECT statement  The CONTAINS clause is specified in the FROM clause of a SELECT statement and works much like the CONTAINS search condition but also returns a score for each matching column, and an overall score for each matching row. See “FROM clause” [SQL Anywhere Server - SQL Reference].

  ○ CREATE TEXT CONFIGURATION statement  This statement creates a text configuration object. A text configuration object is a set of configuration settings that control characteristics of a text index. See “CREATE TEXT CONFIGURATION statement” [SQL Anywhere Server - SQL Reference].

  ○ ALTER TEXT CONFIGURATION statement  This statement alters a text configuration object. See “ALTER TEXT CONFIGURATION statement” [SQL Anywhere Server - SQL Reference].
○ **DROP TEXT CONFIGURATION statement**  This statement drops a text configuration object. See “DROP TEXT CONFIGURATION statement” [SQL Anywhere Server - SQL Reference].

○ **CREATE TEXT INDEX statement**  This statement creates a text index. A text index stores complete positional information for every instance of every term in every indexed column. See “CREATE TEXT INDEX statement” [SQL Anywhere Server - SQL Reference].

○ **ALTER TEXT INDEX statement**  This statement alters a text index. See “ALTER TEXT INDEX statement” [SQL Anywhere Server - SQL Reference].

○ **DROP TEXT INDEX statement**  This statement removes a text index from the database. See “DROP TEXT INDEX statement” [SQL Anywhere Server - SQL Reference].

○ **REFRESH TEXT INDEX statement**  This statement refreshes a text index. See “REFRESH TEXT INDEX statement” [SQL Anywhere Server - SQL Reference].

○ **TRUNCATE TEXT INDEX statement**  This statement truncates the data from a text index. See “TRUNCATE TEXT INDEX statement” [SQL Anywhere Server - SQL Reference].

- **ALTER EVENT statement enhancement**  You can now hide the definition for an event handler using the ALTER EVENT...SET HIDDEN statement. This statement results in the obfuscation of the event handler definition stored in the action column of the ISYSEVENT system table. See “ALTER EVENT statement” [SQL Anywhere Server - SQL Reference].

- **BEGIN SNAPSHOT statement**  The BEGIN SNAPSHOT statement lets you control when a snapshot starts for snapshot isolation. See “BEGIN SNAPSHOT statement” [SQL Anywhere Server - SQL Reference].

- **CASE statement and CASE expression enhancements**  For improved compatibility, CASE statements and CASE expressions are now permitted to end with either END or END CASE. See “CASE statement” [SQL Anywhere Server - SQL Reference] and “CASE expressions” [SQL Anywhere Server - SQL Reference].

- **COMMENT statement enhancements**  You can now add comments to the login policies table and to dbspaces. See:
  - “COMMENT statement” [SQL Anywhere Server - SQL Reference]
  - “Login policies” [SQL Anywhere Server - Database Administration]
  - “Additional dbspaces considerations” [SQL Anywhere Server - Database Administration]

- **CREATE MATERIALIZED VIEW statement enhancement**  You can now create a materialized view that is refreshed whenever underlying data changes, using the new IMMEDIATE REFRESH clause of the CREATE MATERIALIZED VIEW statement. See “CREATE MATERIALIZED VIEW statement” [SQL Anywhere Server - SQL Reference].

- **DESCRIBE statement enhancement**  The Interactive SQL DESCRIBE statement now allows you to obtain information about the database or database server that Interactive SQL is connected to. See “DESCRIBE statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference].
• **IF statement and IF expression enhancements**  For improved compatibility, IF statements and IF expressions are now permitted to end with either ENDIF or END IF. See “IF statement” [SQL Anywhere Server - SQL Reference] and “IF expressions” [SQL Anywhere Server - SQL Reference].

• **LOAD TABLE statement enhancements**  When using the LOAD TABLE statement, you can now specify whether the data in the input file is compressed, and/or encrypted, using the new COMPRESSED or ENCRYPTED clauses. See “LOAD TABLE statement” [SQL Anywhere Server - SQL Reference].

• **SELECT statement enhancements**
  - **Enhancements to INDEX clause**  When specifying an index hint using the INDEX clause, you can now specify up to four indexes that the database server must use. See “FROM clause” [SQL Anywhere Server - SQL Reference].
  - **New INDEX ONLY clause**  When specifying an index hint using the INDEX clause, you can optionally specify the INDEX ONLY clause to control whether the database server performs an index-only retrieval (that is, only index data is used to satisfy the query). See “FROM clause” [SQL Anywhere Server - SQL Reference].
  - **New CROSS APPLY and OUTER APPLY clauses**  The SELECT statement has been extended to support apply expressions (specifically, the CROSS APPLY and OUTER APPLY clauses) in the FROM clause. An apply expression is an easy way to specify joins where the right side is dependent upon the left. For example, you can use an apply expression to evaluate a procedure or derived table once for each row in a table expression. See “Joins resulting from apply expressions” [SQL Anywhere Server - SQL Usage] and “FROM clause” [SQL Anywhere Server - SQL Reference].
  - **New OPENSTRING clause**  Using the new OPENSTRING clause, you can now use a SELECT statement to query data in a file. See “FROM clause” [SQL Anywhere Server - SQL Reference].

• **Specify an owner when creating, altering, dropping, or commenting on events**  The CREATE EVENT, ALTER EVENT, DROP EVENT, and COMMENT ON EVENT statements now allow you to optionally specify the owner. See:
  - “CREATE EVENT statement” [SQL Anywhere Server - SQL Reference]
  - “ALTER EVENT statement” [SQL Anywhere Server - SQL Reference]
  - “DROP EVENT statement” [SQL Anywhere Server - SQL Reference]
  - “COMMENT statement” [SQL Anywhere Server - SQL Reference]

• **UNLOAD statement enhancements**  When using the UNLOAD statement, you can now specify whether to compress and/or encrypt the data that is being unloaded by specifying the COMPRESSED or ENCRYPTED clauses, respectively. See “UNLOAD statement” [SQL Anywhere Server - SQL Reference].

Files compressed or encrypted using these clauses can only be loaded (for example, using LOAD TABLE) by SQL Anywhere 11.0.0 database servers. Files compressed or encrypted using other tools are not usable by SQL Anywhere.
● **UPDATE statement enhancement** For both search and positioned updates, you can now use the SET clause to set the column value to its default value. See “UPDATE statement” [SQL Anywhere Server - SQL Reference] and “UPDATE (positioned) statement [ESQL [SP]]” [SQL Anywhere Server - SQL Reference].

● **Extension to the OPTION clause** The OPTION clause for the INSERT, UPDATE, DELETE, SELECT, UNION, EXCEPT, and INTERSECT statements can now override the setting of the user_estimates database option. See:
  - “INSERT statement” [SQL Anywhere Server - SQL Reference]
  - “UPDATE statement” [SQL Anywhere Server - SQL Reference]
  - “DELETE statement” [SQL Anywhere Server - SQL Reference]
  - “SELECT statement” [SQL Anywhere Server - SQL Reference]
  - “UNION statement” [SQL Anywhere Server - SQL Reference]
  - “EXCEPT statement” [SQL Anywhere Server - SQL Reference]
  - “INTERSECT statement” [SQL Anywhere Server - SQL Reference]

### Programming interfaces

Following is a list of enhancements to programming interfaces introduced in SQL Anywhere version 11.0.0.

● **New SQL Anywhere C API** The SQL Anywhere C application programming interface (API) simplifies the creation of C and C++ wrapper drivers for several interpreted programming languages, including PHP, Perl, Python, and Ruby. The SQL Anywhere C API is layered on top of the DBLIB library and it was implemented with Embedded SQL.

  Although it is not a replacement for DBLIB, the SQL Anywhere C API simplifies the creation of applications using C and C++. You do not need an advanced knowledge of embedded SQL to use the SQL Anywhere C API. See “SQL Anywhere C API support” [SQL Anywhere Server - Programming].

● **New Python Database API (sqlanydb)** The new Python Database API (sqlanydb) provides access to SQL Anywhere databases from scripts written in Python. The sqlanydb module implements, with extensions, the Python Database API specification, version 2.0. See “Python support” [SQL Anywhere Server - Programming].

● **External environments** SQL Anywhere now includes support for six external runtime environments: Java, Perl, PHP, CLR, embedded SQL, and ODBC. SQL Anywhere has had the ability to call compiled native functions written in C or C++ for some time. However, when these procedures are run by the server, the dynamic link library or shared object has always been loaded by the database server and the calls out to the native functions have always been made by the database server. The risk here is that if the native function causes a fault, then the database server will crash. Running compiled native functions outside the database server, in an external environment, eliminates these risks to the server. See “SQL Anywhere external environment support” [SQL Anywhere Server - Programming].

A database upgrade is required to take advantage of this new feature. See “SQL Anywhere Server upgrades” on page 272.
- **PHP external environment support**  SQL Anywhere 11.0.0 includes a variety of pre-built binaries for various PHP versions including 5.1.1 through 5.1.6 and 5.2.0 through 5.2.6. If you have any one of these versions already installed on your server computer, then you should use the SQL Anywhere pre-built binaries instead of building the PHP external environment. Note that, for Linux and Solaris, both 32-bit and 64-bit versions of these binaries are provided. For Windows and other systems, only 32-bit versions are provided.

If you have a different PHP version installed than the ones listed above, then you must build the software, or switch your PHP version to one that matches a SQL Anywhere prebuilt version. For instructions on building the SQL Anywhere PHP module, see “SQL Anywhere PHP extension” [SQL Anywhere Server - Programming].

- **Perl external environment support**  It is very important that you update your version of the SQL Anywhere Perl DBD driver before you try to use the Perl external environment. If you do not update your Perl DBD driver, then server-side Perl will not work.

Also, unlike PHP, SQL Anywhere does not include pre-built binaries for various versions of Perl. Source code for the SQL Anywhere Perl DBD driver is located in `%SQLANY11%SDK/perl`. For instructions on building the SQL Anywhere Perl DBD driver, see “Perl DBI support” [SQL Anywhere Server - Programming].

- **Web server support for UTF-8 URLs**  Previously, the web server decoded percent-encoded (%encoded) data within the request URL (or application/x-www-form-urlencoded data within the body of the request) into the database character set. Now, the contents of percent-encoded (%encoded) data is tested for UTF-8 sequences and converted to the database character set on a maximal extent basis. Any encoded data that is not UTF-8 is decoded and treated as if it is already in the database character set.

Client HTTP applications should send percent-encoded (%encoded) UTF-8 data exclusively. Note that ASCII is represented in UTF-8 as is. For example, a space is encoded as %20.

- **New client callback API**  A new client callback API has been added in support of the new client-side loading and unloading of data features. For embedded SQL, see `DB_CALLBACK_VALIDATE_FILE_TRANSFER` in “`db_register_a_callback` function” [SQL Anywhere Server - Programming]. For ODBC, see `SA_REGISTER_VALIDATE_FILE_TRANSFER_CALLBACK` in “SQLSetConnectAttr extended connection attributes” [SQL Anywhere Server - Programming].

- **SQL_ATTR_CONNECTION_DEAD promptly detects dead connection**  Using ODBC’s `SQLGetConnectAttr` call to get the `SQL_ATTR_CONNECTION_DEAD` attribute now gets the value `SQL_CD_TRUE` if the connection has been dropped even if no request has been made to the server since the connection was dropped. Determining if the connection has been dropped is done without making a request to the server, and the dropped connection is detected within a few seconds. The connection can be dropped for several reasons, for example, on an idle timeout. Before this change, `SQL_ATTR_CONNECTION_DEAD` only got the value `SQL_CD_TRUE` if the connection was disconnected or if ODBC driver made a request to the server (by calling `SQLExecDirect` for example) after the connection was dropped. See “How to get connection attributes” [SQL Anywhere Server - Programming].
● **JDBC Driver now supports ResultSet.getBlob().getBinaryStream()**  The iAnywhere JDBC Driver currently supports the ResultSet.getBlob() method even though this method is optional in the JDBC specification. Support has been added for the optional ResultSet.getBlob().getBinaryStream() method. See “JDBC 4.0 API support” [SQL Anywhere Server - Programming].

● **iAnywhere JDBC driver now accepts jdbc:ianywhere as URL header in addition to jdbc:odbc**  Previously, applications using the URL header jdbc:odbc could be reasonably certain that the JDBC Driver Manager would use the iAnywhere JDBC driver for making connections using this URL. However, recent versions of the Java VM have started to register the Sun JDBC-ODBC bridge as a JDBC driver, and since the Sun JDBC-ODBC bridge also accepts URLs beginning with jdbc:odbc, the chance of an application getting the Sun JDBC-ODBC bridge instead of the iAnywhere JDBC driver is quite high. To guarantee that the JDBC Driver Manager uses the iAnywhere JDBC driver instead of the Sun JDBC-ODBC bridge, the application should use the URL header jdbc:ianywhere instead. See “Connections from a JDBC client application” [SQL Anywhere Server - Programming].

● **ODBC driver manager now accepts driver=iAnywhere Solutions 11 - Oracle**  The Unix ODBC driver manager now accepts driver=iAnywhere Solutions 11 - Oracle, and it loads the threaded iAnywhere ODBC driver for Oracle if the application is threaded. It does not load the driver if the application is non-threaded because the non-threaded iAnywhere ODBC driver for Oracle is not supported. See “SQL Anywhere 16 - Oracle ODBC driver” [MobiLink - Server Administration].

● **ODBC driver manager now accepts driver=UltraLite 11**  The Unix ODBC driver manager already accepts driver=SQL Anywhere 10 and loads the SQL Anywhere ODBC driver (either threaded or non-threaded, depending on the application). The Unix ODBC driver manager now accepts driver=SQL Anywhere 11 and driver=UltraLite 11. For the UltraLite driver, the driver manager only loads the threaded version of the UltraLite ODBC driver because only the threaded version exists.

● **TDS connections enhancement**  The SQL Anywhere database server now allows TDS connections to the default database, even when the Open Client login server name does not match the name of the default database, if the connection string does not involve starting a database (that is, there is no DBF=... ) and if the database server is only running one database.

● **Administration Tool launchers now easier to redeploy**  The launcher executables for the database tools (Sybase Central, DBISQL, DBConsole, ML Monitor) are now easier to redeploy. Registry entries and a set directory structure for the location of the JAR files are no longer required. Each executable needs to have an .ini file in the same directory (with the same name as the executable file) containing the details on how to load the tool. See “Administration tool deployment” [SQL Anywhere Server - Programming].

● **SQL Anywhere .NET Data Provider now supports distributed transaction enlistment**  The .NET 2.0 framework introduced a new namespace System.Transactions, which contains classes for writing transactional applications. Client applications can create and participate in distributed transactions with one or multiple participants. Client applications can implicitly create transactions using the TransactionScope class. The connection object can detect the existence of an ambient transaction created by the TransactionScope and automatically enlist. Client applications can also create a CommittableTransaction and call the EnlistTransaction method to enlist.
This feature is supported by the SQL Anywhere .NET Data Provider. Distributed transaction has significant performance overhead. It is recommended that you use database transactions for non-distributed transactions. See “Transaction processing” [SQL Anywhere Server - Programming].

- **SQL Anywhere .NET Data Provider now supports named parameters**  
  The SQL Anywhere provider now supports named parameters in SACommand. If the user specifies all parameter names, the provider maps the parameter values when the command is executed. When you use named parameters, the order of parameters is not required to match the order of host variables.

  ```csharp
  SACommand cmd = new SACommand(
      "UPDATE MyTable SET name = :name WHERE id = :id", conn);
  
  SAParameter p1 = new SAParameter(
      "id", SADBType.Integer);
  p1.Direction = ParameterDirection.Input;
  p1.Value = 1;
  cmd.Parameters.Add(p1);
  
  SAParameter p2 = new SAParameter(
      "name", SADBType.Char, 40);
  p2.Direction = ParameterDirection.Input;
  p2.Value = "asdasd";
  cmd.Parameters.Add(p2);
  
  cmd.ExecuteNonQuery();
  ```

- **Web services enhancements**  
The following web services enhancements have been made in this release:

  - **Extending web client service procedures of type HTTP:POST to allow a user-defined body**  
The TYPE clause of the CREATE PROCEDURE and CREATE FUNCTION statements has been extended to allow the specification of a mime type. See “CREATE FUNCTION statement [Web service]” [SQL Anywhere Server - SQL Reference] or “CREATE PROCEDURE statement [Web service]” [SQL Anywhere Server - SQL Reference].

  - **Extending web service client procedures to support the PUT, DELETE, and HEAD HTTP methods**  
  Web service client procedures and functions now support the PUT, DELETE and HEAD HTTP methods. The TYPE clause of the CREATE PROCEDURE and CREATE FUNCTION statements has been extended to support these methods. Similar to the POST method, PUT requires a content-type extension within the type clause and only a single (non-substitution) parameter is permitted. See “CREATE SERVICE statement [HTTP web service]” [SQL Anywhere Server - SQL Reference], “CREATE SERVICE statement [SOAP web service]” [SQL Anywhere Server - SQL Reference], “CREATE FUNCTION statement [Web service]” [SQL Anywhere Server - SQL Reference], and “CREATE PROCEDURE statement [Web service]” [SQL Anywhere Server - SQL Reference].

  - **sa_http_php_page and sa_http_php_page_interpreted system procedures**  
The new web service system procedures sa_http_php_page and sa_http_php_page_interpreted return the result of passing a PHP script through a PHP interpreter. See “sa_http_php_page system procedure” [SQL Anywhere Server - SQL Reference] and “sa_http_php_page_interpreted system procedure” [SQL Anywhere Server - SQL Reference].
- **HTTP_BODY system function** A new web service function has been added. The HTTP_BODY function returns the body of the HTTP request in binary form. See “HTTP_BODY function [Web service]” [SQL Anywhere Server - SQL Reference].

- **WSDL support for generating web service client SOAP procedures** WSDLLC now supports the generation of SQL SOAP (web service) client procedures for SQL Anywhere. WSDLLC reads a WSDL1.1 compliant URL or file and generates procedures (or functions) with appropriate parameters and clauses that map to respective SOAP operations listed within the WSDL. The generated SQL statements are written to a SQL file.

- **HTTP SOAP services defined with a FORMAT 'CONCRETE' clause may be further qualified with EXPLICIT OFF or ON** When creating an HTTP SOAP service, the default for the FORMAT clause is EXPLICIT ON. This means that the WSDL generated by a DISH service specifies explicit names and data types for each column returned within a result set. This allows SOAP client toolkits to automatically generate client-side objects and interfaces that represent the result set providing native access to the column values. Before this feature, column values could only be accessed as abstract XML data elements. That behavior can still be achieved by specifying EXPLICIT OFF.

  For more information about how to define an EXPLICIT response object or the generic SimpleDataset, see “CREATE SERVICE statement [SOAP web service]” [SQL Anywhere Server - SQL Reference] and “Tutorial: Using JAX-WS to access a SOAP/DISH web service” [SQL Anywhere Server - Programming].

- **Support for JSON web services** SQL Anywhere now supports web services that return JSON-formatted responses. See “CREATE SERVICE statement [HTTP web service]” [SQL Anywhere Server - SQL Reference].

- **Logging web service clients** The database server now supports logging web service client connections to an output file. You can specify the -zoc server option or use the WebClientLogFile and WebClientLogging properties with the sa_server_option system procedure to control logging and specify the location of the web service client log file. You can also disable the use of this feature with the -sf server option. See:
  
  - “-zoc database server option” [SQL Anywhere Server - Database Administration]
  - “-sf database server option” [SQL Anywhere Server - Database Administration]
  - “sa_server_option system procedure” [SQL Anywhere Server - SQL Reference]

---

**Windows Mobile enhancements**

Following is a list of Windows Mobile enhancements introduced in SQL Anywhere version 11.0.0.

- **-gss server option supported** On Windows CE 4 (Pocket PC 2003) and later, you can use the -gss server option to specify the default stack size for internal execution threads. See “-gss database server option” [SQL Anywhere Server - Database Administration].

- **Checksums enabled by default** When a database is running on Windows Mobile, the database server enables checksums automatically, regardless of whether checksums were enabled for the
database. You must upgrade existing databases or create a new database to use this feature. See “Corruption detection using checksums” [SQL Anywhere Server - Database Administration].

Unix/Linux enhancements

Following is a list of Unix and Linux enhancements introduced in SQL Anywhere version 11.0.0.

- **Controlling the permissions for temporary files** In previous releases, temporary files created by the database server and client were created with global read, write and execute permissions. You can control the permissions for temporary files by setting the SATMP environment variable to a directory with the desired permissions. See “SATMP environment variable” [SQL Anywhere Server - Database Administration].

- **SELinux support** SELinux policies control an application's access to system resources. You can use the default policy on Red Hat Enterprise Linux 5 with SQL Anywhere, but SQL Anywhere is not secured when it is run this way. SQL Anywhere now includes a policy that secures it on Red Hat Enterprise Linux 5. You must compile and install the policy for it to work. The policy source code is provided as part of your SQL Anywhere installation.

  For information about compiling and installing the SQL Anywhere SELinux policy, see $SQLANY11/selinux/readme.

- **Applications menu items on Linux** When installing SQL Anywhere 11 on Linux, you can choose to create Applications menu items.

Mac OS X enhancements

- **Encryption now supported on Mac OS X** RSA communication encryption is now supported by both the database server and clients on Mac OS X. For information about using strong encryption, see “Transport-layer security” [SQL Anywhere Server - Database Administration].

- **HTTPS now supported on Mac OS X** HTTPS communications are now supported by the database server on Mac OS X. For information about using HTTPS, see “-xs database server option” [SQL Anywhere Server - Database Administration].

Miscellaneous

Following is a list of miscellaneous enhancements introduced in SQL Anywhere version 11.0.0.

- **Altering invalid views** Previously a regular view with INVALID status could not be altered, requiring you to drop the view and recreate it. Now, you can alter an invalid view to change the definition so that it is no longer invalid.

- **Synonym added for the LENGTH function** You can now use LEN as a synonym for LENGTH when executing the LENGTH function. See “LENGTH function [String]” [SQL Anywhere Server - SQL Reference].

- **Support added for big-endian and little endian UTF-16 encodings** SQL Anywhere now supports both big-endian and little endian UTF-16 encoding on all platforms, regardless of the
endianness of the platform. You can use UTF-16 encoding in the LOAD TABLE and UNLOAD statements and with the CSCONVERT function. However, you cannot use UTF-16 encoding as the encoding for a connection or database. See “LOAD TABLE statement” [SQL Anywhere Server - SQL Reference] and “UNLOAD statement” [SQL Anywhere Server - SQL Reference].

A database upgrade is required to take advantage of this feature. See “SQL Anywhere Server upgrades” on page 272.

- **Index performance enhancements**  Index performance has been improved, especially when you are operating with a full cache. To benefit from the index performance enhancements, you must rebuild your indexes. The easiest way to do this is to rebuild the database. After rebuilding, you may find that your database file is much smaller. This is normal and should not be a cause for concern.

- **INLINE and PREFIX settings now respected for compressed columns**  Previously, the INLINE and PREFIX settings specified for a column were ignored and treated as 0 if the column was compressed. Now, the settings for the column are respected, even if the column is compressed. See “BLOB considerations” [SQL Anywhere Server - Database Administration] and “CREATE TABLE statement” [SQL Anywhere Server - SQL Reference].

- **Host variables now allowed in batches**  References to host variables are now allowed within batches, with some restrictions. See “Batches” [SQL Anywhere Server - SQL Usage].

- **Enhancements to the InList algorithm**  Previously, the InList algorithm was used by the optimizer only if all the elements of the IN list were either constant values or could be evaluated at optimization time to a constant value. Now, the IN list predicate can contain values that are evaluated only at open time (such as CURRENT DATE, CURRENT TIMESTAMP, or non-deterministic system and user-defined functions), as well as values that are constant within one execution of a query block (outer references).

- **Plan caching for simple DML statements**  Plan caching has been extended to include SELECT statements that qualify for query bypass (simple statements). See “Plan caching” [SQL Anywhere Server - SQL Usage].

- **Size of new databases reduced**  The following system table columns are now compressed to reduce the size of new (empty) databases by approximately 200 KB. This is beneficial when creating databases for use on Windows Mobile.
  
  - ISYSEVENT.action
  - ISYSJARCOLUMN.COMPARTMENT.contents
  - ISYSPROCEDURE.proc_defn
  - ISYSSOURCE.source
  - ISYSTEXTCONFIG.char_stoplist
  - ISYSTEXTCONFIG.nchar_stoplist
  - ISYSTRIGGER.trigger_defn
  - ISYSVIEW.view_def

- **Increased default and minimum packet size**  The default packet size has been increased to 7300 bytes on all operating systems except Windows Mobile. On Windows Mobile, the default continues to be 1460 bytes. The minimum packet size has been increased to 500 bytes. See “CommBufferSize”
● **New ODBC classes supported for remote data access**  Support for the following ODBC classes has been added:

○ MSACCESSODBC
○ MYSQLODBC
○ ULODBC
○ ADSODBC

For more information, see “Server classes for remote data access” [SQL Anywhere Server - SQL Usage].

**Note**

If you previously used the SQL Anywhere for MS Access Migration utility (upsize tool) to migrate Microsoft Access databases to SQL Anywhere, you can now use the MSACCESSODBC class.

● **Database server messages enhancements**  Messages from the database server now have a category and severity assigned to them. You can access this information using the sa_server_messages system procedure, and you can configure the number of messages maintained with the MessageCategoryLimit property. See “sa_server_messages system procedure” [SQL Anywhere Server - SQL Reference].

● **New VALIDATE_COMPLETE parameter for a_validate_type enumeration**  The a_validate_type enumeration has a new parameter, VALIDATE_COMPLETE for performing all possible validations on the database. See “a_validate_db structure [database tools]” [SQL Anywhere Server - Programming].

● **External unload enhancements**  When you perform an external unload of a database, the beginning of the *reload.sql* that is generated now contains a commented CREATE DATABASE statement. This statement can be used to create a database that is equivalent to the one being unloaded.

If the unloaded database was created with version 9 or earlier of SQL Anywhere and had a custom collation, the COLLATION clause appears as follows:

```
COLLATION collation-label DEFINITION collation-definition
```

where *collation-definition* is a string that specifies the custom collation.

If the unloaded database was created with strong encryption, the value of the KEY clause in the CREATE DATABASE statement appears as three question marks (???).

For more information, see “Internal versus external unloads and reloads” [SQL Anywhere Server - Database Administration].
• **New SQL Anywhere Extension Agent OIDs** The following OIDs have been added in this release:
  - saAgent.saRestart
  - saAgent.saInifile

  For more information, see “SQL Anywhere MIB reference” [SQL Anywhere Server - Database Administration].

• **Deadlock system event** The Deadlock system event fires whenever a deadlock occurs. The event handler can use the sa_report_deadlocks procedure to obtain information about the conditions that led to the deadlock. You must upgrade existing databases if you want to use the Deadlock system event. See “System events” [SQL Anywhere Server - Database Administration].

• **Increased database limits** Several SQL Anywhere database limits have been increased. See “SQL Anywhere size and number limitations” [SQL Anywhere Server - Database Administration].

• **Changes to execution plans** Long plans generated by the optimizer now display the following entries related to the overall plan:

  - **Costed Best Plan** Number of different best access plans found by the optimizer.
  - **Costed Plans** Number of different access plans considered by the optimizer.
  - **Optimization Time** The time spent optimizing the query.

  See “Execution plan components” [SQL Anywhere Server - SQL Usage].

  Graphical plans now display the following entries:

  - **Costed Best Plan** Located in the Optimizer Statistics section of the root node, this entry provides the number of different best access plans found by the optimizer.
  - **Costed Plans** Located in the Optimizer Statistics section of the root node, this entry provides the number of different access plans considered by the optimizer.
  - **Optimization Time** Located in the Optimizer Statistics section of the root node, this entry provides the time spent optimizing the query.
  - **FirstRowRunTime** Located in any Node Statistics section, this entry provides the time to fetch the first row.
  - **Joins considered** Located in the Advanced Details section of any join operator, this entry lists all join operators considered by the optimizer during the optimization process for the subtree on the right side of the join operator.
  - **Prefilter predicates** Located in a new scan node section in the Details pane, this entry lists all predicates that are evaluated before the scan is started.
  - **Scan predicates** Located in a scan node section in the Details pane, this entry lists the predicates that are evaluated as columns that are fetched from the row. If a scan predicate rejects a
row, further columns are not read. Scan predicates are simple, single column predicates such as \[ T.x \leq 3 \text{ or } T.x \text{ IS NULL}. \]

- **Post scan predicates**  Located in a new scan node section in the **Details** pane, this entry lists the predicates that are evaluated immediately after a row has been read from the table page. Post scan predicates can refer to multiple columns and can use functions or arithmetic.

- **Residual predicates**  Located in a new scan node section in the **Details** pane, this entry lists predicates that are evaluated after a set of rows has been fetched into memory. Residual predicates usually contain complex operations such as subqueries or user-defined functions and cannot be evaluated as scan predicates or post scan predicates.

- **Indexes considered**  Located in the **Advanced Details** pane, this entry lists all the index or table scans considered by the optimizer during the optimization process for the table referenced by this scan operator. The format of each item in the list is similar to the details listed for a scan operator used in the access plan in the **Details** pane.

- **Primary Key Table**  Located in the **Index** section of an index scan operator, this entry provides the primary key table name.

- **Primary Key Table Estimated Rows**  Located in the **Index** section of an index scan operator, this entry provides the number of rows in the primary key table.

- **Primary Key Column**  Located in the **Index** section of an index scan operator, this entry provides the names of the primary key columns.

- **Sequential Transitions**  Located in the **Index** section of an index scan operator, this entry provides the statistics kept for each physical index indicating how clustered the index is.

- **Random Transitions**  Located in the **Index** section of an index scan operator, this entry provides the statistics kept for each physical index indicating how clustered the index is.

- **Key Values**  Located in the **Index** section of an index scan operator, this entry provides the number of unique entries in the index.

### SQL Anywhere behavior changes

Following is a list of behavior changes to SQL Anywhere introduced in version 11.0.0, grouped by category.

- **Catalog changes**  The following table contains the changes to the catalog for version 11.0.0.

  You must upgrade your database to get these changes. See “SQL Anywhere Server upgrades” on page 272.
<table>
<thead>
<tr>
<th>Table name and/or view name</th>
<th>Description of change</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISYSTAB/ SYSTAB</td>
<td>○ A new column, dbspace_id has been added as an eventual replacement to the current file_id column.</td>
</tr>
<tr>
<td></td>
<td>○ The file_id column is deprecated. Use dbspace_id instead. For global temporary tables, SYSTAB.file_id now points to the temporary dbspace, instead of the system dbspace.</td>
</tr>
<tr>
<td></td>
<td>○ A new column, last_modified_tsn, has been added to store a sequence number for the transaction that modified the table.</td>
</tr>
<tr>
<td>ISYSIDX/ SYSIDX</td>
<td>○ A new column, dbspace_id has been added as an eventual replacement to the current file_id column.</td>
</tr>
<tr>
<td></td>
<td>○ The file_id column is deprecated. Use dbspace_id instead.</td>
</tr>
<tr>
<td>ISYSFILE</td>
<td>This system table is deprecated. All columns, with the exception of lob_map, are now found in the (new) ISYSDBSPACE system table. The lob_map column is now found in the (new) ISYSDBFILE system table.</td>
</tr>
<tr>
<td>ISYSDBFILE/ SYSDBFILE</td>
<td>New table to hold information about dbspaces.</td>
</tr>
<tr>
<td>ISYSDBSPACE/ SYSDBSPACE</td>
<td>New table to hold information about dbspaces.</td>
</tr>
<tr>
<td>SYSDBSpace-Perm/ ISYSDBSPACE</td>
<td>New table to hold dbspace permissions.</td>
</tr>
<tr>
<td>ISYSOBJECT/ SYSOBJECT</td>
<td>The file_id column has been renamed to dbspace_id. Also, the object_type column can contain two new values: 17 (Text configuration), and 18 (Dbspace).</td>
</tr>
<tr>
<td>SYSINDEXES</td>
<td>The indextype field now identifies foreign keys and primary key indexes as Primary Key and Foreign Key, respectively to distinguish them from other indexes.</td>
</tr>
<tr>
<td>ISYSCAPABILITY-NAME</td>
<td>This table no longer exists in the catalog. The corresponding SYSCAPABILITYNAME system view is still available, but it generated using server properties.</td>
</tr>
<tr>
<td>ISYSEVENTTYPE</td>
<td>This table no longer exists in the catalog. The corresponding SYSEVENTTYPE system view is still available, but it generated using server properties.</td>
</tr>
<tr>
<td>Table name and/or view name</td>
<td>Description of change</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>ISYSVIEW</td>
<td>New column called mv_last_refreshed_tsn to store a sequence number for the transaction that refreshed the materialized view.</td>
</tr>
<tr>
<td>ISYSLOGINMAP/SYSLOGINMAP</td>
<td>New table to hold information about login policies.</td>
</tr>
<tr>
<td>ISYSLOGINPOLICY/SYSLOGINPOLICY</td>
<td>New table to hold information about login policies.</td>
</tr>
<tr>
<td>ISYSLOGINPOLICYOPTION/ SYSLOGINPOLICYOPTION</td>
<td>New table to hold information about login policies.</td>
</tr>
<tr>
<td>ISYSTEXTCONFIG/SYSTEXTCONFIG</td>
<td>New table to hold information about text configuration objects.</td>
</tr>
<tr>
<td>ISYSTEXTIDX/SYSTEXTIDX</td>
<td>New table to hold information about text indexes.</td>
</tr>
<tr>
<td>ISYSTEXTIDXTAB/SYSTEXTIDXTAB</td>
<td>New table to hold information about text indexes.</td>
</tr>
</tbody>
</table>

- **PHP function name changes**  All PHP functions have been renamed to have sasql_ as their prefix, instead of sqlanywhere_. The sqlanywhere_ prefix is still allowed in the name when calling a function, but is deprecated. You should change your application to use the new prefix.

- **INSERT...ON EXISTING UPDATE statement now fires triggers**  Previously, when you executed an INSERT...ON EXISTING UPDATE statement, triggers did not fire if data was updated. Now, the database server fires statement-level after triggers for the updates.

- **REFRESH MATERIALIZED VIEW statement**  You can no longer specify STATEMENT SNAPSHOT and READONLY STATEMENT SNAPSHOT as the isolation level for the refresh since the effect of these options is the same as specifying SNAPSHOT for the isolation level. See “REFRESH MATERIALIZED VIEW statement” [SQL Anywhere Server - SQL Reference].

- **REORGANIZE TABLE statement**  Attempting to execute multiple REORGANIZE TABLE statements simultaneously on the same table now results in an error.

- **sa_validate system procedure**  The check_type, express, and checksum arguments for sa_validate are now obsolete; specifying them no longer has an effect. Checksum validation is now performed by default. Also, when the sa_validate system procedure is called without specifying any arguments, in addition to validating all tables, materialized views, and indexes, the database server also validates the database itself, including checksums. See “sa_validate system procedure” [SQL Anywhere Server - SQL Reference].
- **-gss server option**  
  The -gss server option is now supported on Windows XP and later. In previous releases, this option was not supported on Windows operating systems. See “-gss database server option” [SQL Anywhere Server - Database Administration].

- **-gx server option no longer supported**  
  Support for the -gx server option has been removed in this release. Specifying the -gx option when starting a SQL Anywhere database server results in an error.

- **LazyClose connection parameter default setting is now AUTO**  
  In previous versions, when an application closed a cursor, a round trip to the database server was required unless the LazyClose connection parameter was set to NO. Now, cursor close requests are queued for many cursors by default, eliminating a round trip and resulting in improved performance. The LazyClose connection parameter now accepts three values: YES, NO, and AUTO (the default). YES was the default setting in previous releases. See “LazyClose (LCLOSE) connection parameter” [SQL Anywhere Server - Database Administration].

- **Embedded SQL import library changes**  
  The Watcom and Borland versions of the DBLIB import libraries are no longer included. These are dblibtw.lib and dblibtb.lib, respectively. An import definition file (%SQLANY11%SDK\lib\Def\dblib.def file) is provided as a replacement for these import libraries.

- **Database tools import library changes**  
  The Watcom and Borland versions of the database tools import libraries are no longer included. These are dbtlstw.lib and dbtlstb.lib, respectively. An import definition file (%SQLANY11%SDK\lib\Def\dbtool.def) is provided as a replacement for these import libraries.

- **DBLIB indicator behavior defined when no rows received**  
  On a fetch or execute where no rows are received from the database server (on an error or the end of the result set), indicator values are now unchanged. See “Indicator variables” [SQL Anywhere Server - Programming].

- **ODBC SQLGetConnectAttr**  
  Using the ODBC SQLGetConnectAttr call to get the SQL_ATTR_CONNECTION_DEAD attribute now gets the value SQL_CD_TRUE if the connection has been dropped, even if no requests have been sent to the server since the connection was dropped. Determining if the connection has been dropped is done without making a request to the server, and the dropped connection is detected within a few seconds. The connection can be dropped for several reasons, such as an idle timeout.

  In previous releases, SQL_ATTR_CONNECTION_DEAD only got the value SQL_CD_TRUE if the connection was disconnected or if the ODBC driver made a request to the server (for example, by calling SQLExecDirect) after the connection was dropped.

- **Databases cannot be created or started that are named utility_db**  
  The name utility_db is now reserved for the SQL Anywhere Server utility database. If you attempt to create a new database or start an existing database named utility_db.db, an error is returned. If you have an existing database named utility_db, you can start it with a different name. See “The utility database” [SQL Anywhere Server - Database Administration].

- **Computed column dependencies**  
  Previously, to allow an update or insert operation to proceed without error, an application could have used triggers to assign non-NULL values to columns that were declared NOT NULL. This impacted computed columns that were dependent on the column,
since it could result in a computed value that did not reflect the intended computation. Now, an attempt to set a NULL value in a NOT NULL column that a computed column depends on, fails with an error message and no triggers are fired. See “Inserts into, and updates of, computed columns” [SQL Anywhere Server - SQL Usage].

- **Dbspace names containing a period generate an error** In previous releases, if a dbspace name that was not quoted contained a period, then the part of the dbspace name before the period was silently ignored by the server. The database server now generates an error for these names.

- **SQL Anywhere web server no longer supports SSL version 2.0** When using the SQL Anywhere web server, only SSL version 3.0 and TLS version 1.0 connections are supported. SSL version 2.0 connections are not supported.

- **CREATE SERVICE option DATATYPE default value has changed** The default value of the DATATYPE clause has changed from OFF to ON. If you want the old behavior then you must explicitly include DATATYPE OFF in the CREATE SERVICE definition. See “CREATE SERVICE statement [SOAP web service]” [SQL Anywhere Server - SQL Reference].

- **Checksum behavior changes** For databases created with version 11 or upgraded to version 11, the database server automatically enables checksums to databases running off of media such as network drives or removable drives. Checksums remain enabled as long as the database resides on such a device. See “Corruption detection using checksums” [SQL Anywhere Server - Database Administration].

- **HTTP connections do not cause databases to stop automatically** In previous releases, when you configured a database to stop automatically, the database would stop if an HTTP connection disconnected and there were no other connections to the database. Databases now stop automatically only when the last command sequence or TDS connection disconnects.

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 does not stop automatically. As well, if a database that is configured to stop automatically has an HTTP connection and a command sequence or TDS connection, when the last command sequence or TDS connection disconnects, the database stops, and any HTTP connections are dropped. See “-ga database server option” [SQL Anywhere Server - Database Administration] and “AutoStop (ASTOP) connection parameter” [SQL Anywhere Server - Database Administration].

- **Database mirroring behavior change** In previous releases, if the connection parameters specified in the -xp option for the primary or mirror server were invalid, the database server would repeatedly attempt to connect, but the connection would never succeed. In this release, 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.

- **Default refresh behavior for materialized views** Previously, the default refresh behavior for materialized views was WITH EXCLUSIVE MODE. Now, default refresh behavior depends on whether the materialized view is defined as IMMEDIATE REFRESH and whether snapshot isolation level is enabled for the database. See “REFRESH MATERIALIZED VIEW statement” [SQL Anywhere Server - SQL Reference].
● **post_login_procedure database option behavior change**  The default setting of the post_login_procedure database option is now the sa_post_login_procedure system procedure. See “post_login_procedure option” [SQL Anywhere Server - Database Administration].

● **non_keywords database option**  In previous releases, in addition to specifying individual keywords, you could also turn off all keywords since a specified release by using the following special values in the list of keywords:

```
keywords_4_0_d, keywords_4_0_c, keywords_4_0_b, keywords_4_0_a,
keywords_4_0,
keywords_5_0_01, keywords_5_0
```

These special values are no longer supported. You can still turn off individual keywords. See “non_keywords option” [SQL Anywhere Server - Database Administration].

● **cooperative_commit_timeout database option**  This option setting is now ignored as commit behavior is automatically tuned.

● **cooperative_commits database option**  This option setting is now ignored as commit behavior is automatically tuned.

● **quoted_identifier database option setting respected for remote data access**  The local setting of the quoted_identifier option now controls the use of quoted identifiers for Adaptive Server Enterprise and Microsoft SQL Server when you are using remote data access. For example, if you set the quoted_identifier option to Off locally, then quoted identifiers are turned off for Adaptive Server Enterprise. See:

  o  “Server class ASEODBC” [SQL Anywhere Server - SQL Usage]
  o  “Server class MSSODBC” [SQL Anywhere Server - SQL Usage]

● **Changes to the scope of the precision and scale database options**  In previous releases, you could set the precision and scale database options for individual users or specify that the setting had a temporary scope. However, these settings can affect the recoverability of a database. If the temporary or user-level settings differ from the corresponding PUBLIC settings when executing DDL statements that create or alter tables and domains, you may encounter problems while rebuilding the database. The following behavior now applies for the precision and scale database options:

<table>
<thead>
<tr>
<th>Database server version</th>
<th>Version 10 or earlier database</th>
<th>Version 11 database</th>
<th>Database upgraded to version 11</th>
<th>Unloading a version 10 or earlier database</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>PUBLIC settings allowed</td>
<td>PUBLIC settings allowed</td>
<td>PUBLIC settings allowed</td>
<td>PUBLIC settings unloaded</td>
</tr>
<tr>
<td></td>
<td>User settings allowed</td>
<td>User settings not allowed</td>
<td>User settings not allowed</td>
<td>User settings discarded during unload</td>
</tr>
<tr>
<td></td>
<td>Temporary settings not allowed</td>
<td>Temporary settings not allowed</td>
<td>Temporary settings not allowed</td>
<td></td>
</tr>
</tbody>
</table>
Version 10 and earlier database servers continue to allow you to set the scale and precision options temporarily, as well as for individual users.

**Caution**
It is recommended that you do not rely on user-level or temporary settings for the precision and scale database options because of the potential problems you can encounter when rebuilding databases, and because of the unpredictable database server behavior that can occur.

See:
- “precision option” [SQL Anywhere Server - Database Administration]
- “scale option” [SQL Anywhere Server - Database Administration]

- **OPTION clause behavior change** The OPTION clause for the INSERT, UPDATE, DELETE, SELECT, UNION, EXCEPT, and INTERSECT statements now returns an error if you specify a database option that the clause does not support. See:
  - “INSERT statement” [SQL Anywhere Server - SQL Reference]
  - “UPDATE statement” [SQL Anywhere Server - SQL Reference]
  - “DELETE statement” [SQL Anywhere Server - SQL Reference]
  - “SELECT statement” [SQL Anywhere Server - SQL Reference]
  - “UNION statement” [SQL Anywhere Server - SQL Reference]
  - “EXCEPT statement” [SQL Anywhere Server - SQL Reference]
  - “INTERSECT statement” [SQL Anywhere Server - SQL Reference]

- **Rollback log behavior change for read-only databases** In previous releases, operations on read-only databases involving transactional temporary objects were not treated as transactional: no rollback log information was kept for them. In this release, transactional temporary objects in read-only databases have fully-transactional semantics. They are subject to commits, rollbacks, and rollbacks to savepoints.

- **Itanium 64-bit supported platform changes** In previous versions, a full 64-bit version of the software was available for Windows Server 2003 on Itanium II chips, and a deployment release was available on 64-bit Linux and HP-UX operating systems.

  In this release, only the deployment release for 64-bit HP-UX is available.
- **Unpack utility (dbunload) behavior changes**  In previous releases, the `dbunload -ea`, `-ek`, and `-ep` options had to be specified with the `-an` or `-ar` option to control encryption for the new database. Now, if you unload a database, or any part of it, but do not reload it, the `-ea`, `-ek`, and `-ep` option control the encryption of the table data files that are created. When you use these files to reload a database from Interactive SQL, you must specify the encryption key as a parameter to the READ statement. See “Unpack utility (dbunload)” [SQL Anywhere Server - Database Administration].

As well, in previous releases the version of `dbunload` used to extract a database did not have to be the same version as the database server running the database. Now, when `dbunload` is used with a version 10.0.0 or later database, the version of `dbunload` used must match the version of the database server used to access the database. If an older version of `dbunload` is used with a newer database server, or vice versa, an error is reported.

- **Extraction utility (dbxtract) behavior change**  In previous releases the version of `dbxtract` used to extract a database did not have to be the same version as the database server running the database. Now, when `dbxtract` is used with a version 10.0.0 or later database, the version of `dbxtract` used must match the version of the database server used to access the database. If an older version of `dbxtract` is used with a newer database server, or vice versa, an error is reported.

- **Changes in locking behavior**  In previous releases, an UPDATE or DELETE statement executing at isolation level 0 could block on a row lock for a row that was not affected by the statement. It is now less likely for an UPDATE or DELETE statement to take an intent or exclusive lock on a row that is not affected by the statement. When developing applications, you should use caution when using isolation level 0 or 1 with UPDATE and DELETE statements, and ensure that the behavior is acceptable for your application. See “Locks during updates” [SQL Anywhere Server - SQL Usage] and “Locks during deletes” [SQL Anywhere Server - SQL Usage].

- **Changes to property names**  The following properties have been renamed in this release:

<table>
<thead>
<tr>
<th>Old name</th>
<th>New name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheHitsEng</td>
<td>CacheHits</td>
</tr>
<tr>
<td>CacheReadEng</td>
<td>CacheRead</td>
</tr>
<tr>
<td>DiskReadEng</td>
<td>DiskRead</td>
</tr>
<tr>
<td>ReadHint</td>
<td>DiskReadHint</td>
</tr>
<tr>
<td>ReadHintScatter</td>
<td>DiskReadHintPages</td>
</tr>
<tr>
<td>ReadHintScatterLimit</td>
<td>DiskReadHintScatterLimit</td>
</tr>
</tbody>
</table>

For more information, see “Connection, database, and database server properties” [SQL Anywhere Server - Database Administration].

- **Language Selection utility (dblang)**  In previous releases, this utility was only installed when the International Resources Development Kit (IRDK) was selected during installation. In this release, all international resources and the Language Selection utility (dblang) are installed all the time.
- **Default dbspace for temporary tables and indexes**  
  Temporary tables can only be created in the TEMPORARY dbspace. If you specify the SYSTEM dbspace in the IN clause of the CREATE TABLE statement, the IN clause is ignored, and the temporary table is created in the temporary dbspace. If you specify a user-defined dbspace in the IN clause of the CREATE TABLE statement, an error is returned. As well, the default_dbspace option is ignored when creating temporary objects.

- **Loading data into temporary tables**  
  When loading data into temporary tables you can no longer load a local temporary table that is ON COMMIT DELETE. In previous releases, you could load data into a local temporary table defined with ON COMMIT DELETE ROWS. An autocommit is now performed automatically when you run a LOAD TABLE statement; in previous releases, this did not always occur.

- **Database server options**  
  The -uc and -ui server options are now supported on Mac OS X. Previously they were only supported on Linux. On Linux the -ui server option opens the **Server Startup Options** window, displays the database server messages window, and starts the database server whether or not the X window server starts. On Mac OS X -ui displays database server messages in a new window and starts the database server in shell mode if a usable display isn't available. The -uc server option starts the database server in shell mode. See “-uc database server option” [SQL Anywhere Server - Database Administration] and “-um database server option” [SQL Anywhere Server - Database Administration].

- **Remote data access no longer works with ODBC drivers that do not support UNICODE calls**  
  Remote data access no longer works with ODBC drivers that do not support UNICODE calls. As a result, with non-UNICODE ODBC drivers, remote data access does not perform any character set translation on data coming in from the ODBC driver.

- **SYSFILE system view**  
  A row for the temporary file is now included in the SYSFILE compatibility view.

---

### SQL Anywhere deprecated and discontinued features

- **FORMAT ASCII clause deprecated for LOAD TABLE, UNLOAD TABLE, INPUT, and OUTPUT statements**  
  The FORMAT ASCII clause for the LOAD TABLE, UNLOAD TABLE, INPUT, and OUTPUT statements has been deprecated and is replaced by FORMAT TEXT. Utilities such as dbunload now generate reload scripts containing FORMAT TEXT rather than FORMAT ASCII.

  For the OUTPUT statement, the FORMAT TEXT clause now writes the data in the same file format as FORMAT ASCII did in previous versions. The output formerly created by FORMAT TEXT is no longer available.

- **Database properties**  
  The following database properties are unsupported:
  
  - MapPages
  - PreserveSource
  - UniqueIdentifier
● **Server properties**  The following server properties have been deprecated in this release:

- MaxMessage
- Message
- MessageTime
- MessageText
- MessageWindowSize

● **SPX protocol unsupported**  Support for the SPX protocol has been removed in this release. As a result, the following protocol options are discontinued:

- ExtendedName protocol option [ENAME]
- RegisterBindery protocol option [REGBIN]
- SearchBindery protocol option [BINSEARCH]

The following features of the SQL Anywhere .NET Data Provider are discontinued:

- SACommLinksOptionsBuilder class: SpxOptionsBuilder property
- SACommLinksOptionsBuilder class: SpxOptionsString property
- SASpxOptionsBuilder class

● **dbinit -e option unsupported**  The dbinit -e option, used for specifying simple encryption when creating a database, is no longer supported. Use the -ea simple option to specify simple encryption. See “Initialization utility (dbinit)” [SQL Anywhere Server - Database Administration].

● **Discontinued database options**  Support for the following database options and their corresponding database properties has been removed in this release.

<table>
<thead>
<tr>
<th>Options</th>
<th>Behavior in this release</th>
</tr>
</thead>
<tbody>
<tr>
<td>ansi_integer_overflow</td>
<td>An overflow now always results in a SQLSTATE = 22003 – overflow error. When unloading, or connecting to, older databases with materialized views, the setting of this option is ignored.</td>
</tr>
<tr>
<td>ansi_substring</td>
<td>The behavior of the SUBSTRING function now corresponds to ANSI/ISO SQL/2003 behavior. See “SUBSTRING function [String]” [SQL Anywhere Server - SQL Reference].</td>
</tr>
<tr>
<td>automatic_timestamp</td>
<td>New columns with the TIMESTAMP data type that do not have an explicit default value defined are never given a default value of the Transact-SQL timestamp.</td>
</tr>
<tr>
<td>divide_by_zero_error</td>
<td>Division by zero now results in an error with SQLSTATE 22012. When unloading or connecting to older databases with materialized views, the setting of this option is ignored.</td>
</tr>
<tr>
<td>Options</td>
<td>Behavior in this release</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| float_as_double       | In SQL Anywhere, the FLOAT keyword never behaves like Adaptive Server Enterprise's FLOAT keyword when a precision is not specified. SQL Anywhere does not treat FLOAT values the same as DOUBLE values.  
For Open Client and jConnect connections, this behavior is different from the default behavior in previous releases.  
When unloading or connecting to older databases with materialized views, the setting of this option is ignored. |
| optimis-tic_wait_for_commit | This option is no longer supported.                                                                                                                                                                                        |
| query_plan_on_open    | A plan is no longer returned when an OPEN is done on a cursor. A more complete description can be obtained using the EXPLAIN statement or the PLAN function. See “EXPLAIN statement [ESQL]” [SQL Anywhere Server - SQL Reference] and “PLAN function [Miscellaneous]” [SQL Anywhere Server - SQL Reference]. |
| ri_trigger_time       | Referential integrity actions are now executed after the UPDATE or DELETE.                                                                                                                                                  |
| truncate_with_auto_commit | A COMMIT is now executed both before and after a TRUNCATE TABLE statement is executed.                                                                                                                                     |
| tsql_hex_constant     | Hexadecimal constants are now treated as binary typed constants.                                                                                                                                                           |
| uuid_has_hyphens      | UUID strings now contain four hyphens.  
When unloading or connecting to older databases with materialized views, the setting of this option is ignored.                                                                                                       |
| percent_as_comment    | In previous releases, the percent sign (%) could be used as a comment marker depending on the setting of the percent_as_comment database option. Now, SQL Anywhere treats the % sign as a modulo operator.  
See “MOD function [Numeric]” [SQL Anywhere Server - SQL Reference]. |

- **SQLANYSH10 environment variable unsupported**  
In previous releases, some of the SQL Anywhere software was installed into a shared directory. This location could be specified by the SQLANYSH10 environment variable. Software is no longer installed into a shared directory as part of the installation process, and the SQLANYSH10 environment variable is no longer used.  

When creating a silent install, you no longer need to set the SHARED_DIR location. See “Silent installs using the SQL Anywhere installer” [SQL Anywhere Server - Programming].
- **sa_get_server_messages system procedure discontinued**  In previous releases, you could use the sa_get_server_messages system procedure to return constants from the database server messages window as a result set. You can now use the sa_server_messages system procedure to obtain this information. See “sa_server_messages system procedure” [SQL Anywhere Server - SQL Reference].

- **background_priority option deprecated**  The background_priority option has been deprecated. Use the priority option instead. See “priority option” [SQL Anywhere Server - Database Administration].

- **encrypt_aes_random_iv option unsupported**  Support for the encrypt_aes_random_iv database option has been removed in this release. Now, a random IV (initialization vector) is always used.

- **DLL protocol option unsupported**  Support for the DLL protocol option has been removed. Windows database servers and clients use Winsock 2.2. Windows Mobile clients use Winsock 1.1. See “TCP/IP protocol” [SQL Anywhere Server - Database Administration].

- **SQL Anywhere Broadcast Repeater utility renamed**  In version 10, the command to run the SQL Anywhere Broadcast Repeater utility was dbns10. In this release, it is dbns11. See “Broadcast Repeater utility (dbns16)” [SQL Anywhere Server - Database Administration].

- **SQLANY10 and SQLANYSAMP10 environment variables renamed**  The SQLANY10 and SQLANYSAMP10 environment variables have been renamed SQLANY11 and SQLANYSAMP11, respectively. See:
  - “SQLANY16 environment variable” [SQL Anywhere Server - Database Administration]
  - “SQLANYSAMP16 environment variable” [SQL Anywhere Server - Database Administration]

- **Some secure features renamed**  The following secure features have been renamed for this release:

<table>
<thead>
<tr>
<th>Deprecated name</th>
<th>New name</th>
</tr>
</thead>
<tbody>
<tr>
<td>xp_read_file</td>
<td>read_file</td>
</tr>
<tr>
<td>xp_write_file</td>
<td>write_file</td>
</tr>
<tr>
<td>unload_table</td>
<td>write_file</td>
</tr>
<tr>
<td>load_table</td>
<td>read_file</td>
</tr>
</tbody>
</table>

For more information, see “-sf database server option” [SQL Anywhere Server - Database Administration].

---

**MobiLink**

The following sections describe the new features, behavior changes, and deprecated features in MobiLink for version 11.0.0.
MobiLink new features

Following is a list of additions to MobiLink introduced in version 11.0.0.

Consolidated databases

- **DB2 mainframe now supported as a consolidated database** MobiLink has long supported DB2 LUW (Linux, Unix, and Windows) as a consolidated database. Now it also supports DB2 mainframe.

- **MySQL now supported as a consolidated database** MobiLink now supports MySQL as a consolidated database.

  See “MySQL consolidated database” [MobiLink - Server Administration].

- **MobiLink System Database (MLSD)** Support has been added for a separate database to hold MobiLink system data (MLSD - MobiLink System Database). This feature must be used with Microsoft DTC (Distributed Transaction Coordinator). See “-cs mlsrv16 option” [MobiLink - Server Administration].

New system objects

- **New MobiLink server system tables and schema** Following are changes to the MobiLink system tables:

  ○ Several new MobiLink system tables have been added:

    - ml_qa_delivery_archive
    - ml_qa_repository_archive
    - ml_qa_repository_props_archive
    - ml_qa_status_history_archive
    - ml_server
    - ml_active_remote_id
    - ml_passthrough
    - ml_passthrough_repair
    - ml_passthrough_script
    - ml_passthrough_status

  ○ Several new MobiLink system procedures have been added. See:

    - “ml_add_passthrough system procedure” [MobiLink - Server Administration]
    - “ml_add_passthrough_repair system procedure” [MobiLink - Server Administration]
    - “ml_add_passthrough_script system procedure” [MobiLink - Server Administration]
    - “ml_delete_passthrough system procedure” [MobiLink - Server Administration]
    - “ml_delete_passthrough_repair system procedure” [MobiLink - Server Administration]
    - “ml_delete_passthrough_script system procedure” [MobiLink - Server Administration]

iAnywhere Solutions Oracle driver

- **Oracle DSN can store an encrypted password** When creating an Oracle ODBC data source in the Windows ODBC Administrator, you can now choose to encrypt the password that is stored in the
ODBC data source. See “SQL Anywhere 16 - Oracle ODBC driver” [MobiLink - Server Administration].

- **Oracle ODBC driver supports Microsoft distributed transactions** The Oracle ODBC driver now supports Microsoft distributed transactions. In the Windows ODBC Administrator, select Enable Microsoft distributed transactions and make sure that the appropriate DLL is installed with the Oracle client. See “SQL Anywhere 16 - Oracle ODBC driver” [MobiLink - Server Administration].

**MobiLink server**

- **Relay server** The relay server is a set of web extensions that enable secure, load-balanced communication between mobile devices and MobiLink, Afaria and OneBridge servers communicating through a web server. See “Introduction to the Relay Server” [Relay Server].

- **Sybase relay server hosting service** The Sybase relay server hosting service is a farm of relay servers hosted by Sybase. It is intended to ease the development of mobile applications that use MobiLink data synchronization and to simplify the evaluation process for developers, especially where data is sent using public wireless networks. See “Sybase Hosted Relay Service” [Relay Server].

- **MobiLink Server Farm** MobiLink servers may now be explicitly grouped into server farms of identical servers.

  Redundant, concurrent synchronizations from the same remote ID are now automatically detected across the entire farm. This removes the need for a load balancer to keep sending the same remote ID to the same MobiLink server.

  You can now configure each MobiLink server identically.

  Failover is automatically supported for the Notifier. The farm automatically appoints a MobiLink server to run these, and will elect a new server to run them if the first appointee computer fails.

- **64-bit Platforms** MobiLink server is now fully 64-bit on several 64-bit platforms. For a list of the supported platforms 64-bit platforms, see http://www.sybase.com/detail?id=1002288.

- **New member for Java DownloadTableData interface** `getLastDownloadTime` method to return last download time for a table. See “DownloadTableData.getLastDownloadTime method [MobiLink server Java]” [MobiLink - Server Administration].

- **SQL passthrough** The SQL Passthrough feature allows you to download scripts of SQL statements from a consolidated database to a SQL Anywhere or UltraLite client, and have those SQL statements executed on the client at an appropriate time.

- **Info message listening** The Java and .NET APIs now allow users to register to receive notifications whenever an info line prefixed with "I" is printed to the log.
See:

- “LogMessage.INFO variable [MobiLink server Java]” [MobiLink - Server Administration]
- “ServerContext.addInfoListener method [MobiLink server Java]” [MobiLink - Server Administration]
- “ServerContext.removeInfoListener method [MobiLink server Java]” [MobiLink - Server Administration]
- “LogMessage.MessageType enumeration [MobiLink server .NET]” [MobiLink - Server Administration]
- “ServerContext.InfoListener event [MobiLink server .NET]” [MobiLink - Server Administration]

New mlsrv11 features

- **-cs option**  MobiLink server system objects such as system tables, procedures, triggers, and views can now be stored in a database other than the consolidated database. The database that stores the MobiLink system objects is called the MobiLink System Database or MLSD.

  Use -cs to specify connection parameters for your MLSD. See “-cs mlsrv16 option” [MobiLink - Server Administration].

- **-lsc option**  Specifies the local server connect information. This information is passed to other servers in the server farm. See “-lsc mlsrv16 option” [MobiLink - Server Administration].

- **-ss option**  Enables the MobiLink server to run in a server farm.

  **Note**
  The -ss option is a feature of the MobiLink high availability option, which requires a separate license. See “Separately licensed components” [SQL Anywhere 16 - Introduction].

- **-tc option**  Set the count down timer for SQL script execution. See “-tc mlsrv16 option” [MobiLink - Server Administration].

- **-tf option**  Fail the SQL script execution when the count down timer specified with -tc expires (not for Oracle). See “-tf mlsrv16 option” [MobiLink - Server Administration].

New MobiLink scripting features

- **Non-blocking download ACK scripts**  See “nonblocking_download_ack connection event” [MobiLink - Server Administration] and “publication_nonblocking_download_ack connection event” [MobiLink - Server Administration].

MobiLink Redirector enhancements

- **Redirector deprecated**  The Redirector is deprecated. It has been replaced by the Relay Server. See “Introduction to the Relay Server” [Relay Server].

- **New Relay Server**  The relay server is a set of web extensions that enable secure, load-balanced communication between mobile devices and MobiLink, Afaria, and OneBridge servers communicating through a web server. See “Introduction to the Relay Server” [Relay Server].
MobiLink clients

SQL Anywhere clients

- dbmlsync APIs for C++ and .NET  The Dbmlsync API provides a programming interface that allows MobiLink client applications written in C++ or .NET to launch synchronizations and receive feedback about the progress of the synchronizations they request. See “Dbmlsync API” [MobiLink - Client Administration].

- Synchronization profiles  You can now store dbmlsync command lines in your database as synchronization profiles. The -sp dbmlsync option enables you to add options from a synchronization profile to command line synchronization options. See “-sp dbmlsync option” [MobiLink - Client Administration].

- LOAD TABLE now optionally logs loaded rows as inserts  The new clause WITH ROW LOGGING means that SQL Anywhere remote databases can now load data using the LOAD TABLE statement. Previously, LOAD TABLE was not always useful in a synchronization environment because the rows were not logged and so they were ignored by dbmlsync. See “Data import with the LOAD TABLE statement” [SQL Anywhere Server - SQL Usage].

- dbmlsync log scanning optimization  Now optimized for non-overlapping publications.

Security

The following security features have been added in this release:

- End-to-end encryption  MobiLink synchronization streams and clients now support protocol-level end-to-end encryption. See “End-to-end encryption” [SQL Anywhere Server - Database Administration].

- Key Pair Generator utility (createkey)  This utility creates RSA and ECC key pairs for use with MobiLink end-to-end encryption. See “Key Pair Generator utility (createkey)” [SQL Anywhere Server - Database Administration].

Server-initiated synchronization

- Support for MobiLink server farm  SIS has been enhanced to operate better in a MobiLink server farm environment. You can now run a Notifier on every MobiLink server in the farm and the Notifiers together ensure that there are no redundant notifications to the same Listener. See “-isc mlsrv16 option” [MobiLink - Server Administration].

Listener enhancements

- New dblsn options  The following options have been added to the MobiLink Listener for Windows:

  - **-ni**  Disable IP tracking.

  - **-pc**  Disable persistent connection.
MobiLink behavior changes and deprecated features

Following is a list of changes to MobiLink introduced in version 11.0.0.

MobiLink server changes

- Change to default download acknowledgement
  The default value for the -nba option is now -nba+, which makes non-blocking download acknowledgement the default. If you use download blocking download acknowledgement in an existing deployment, you must either:
  - Use the -nba- option.
  - Look at your synchronization scripts to use the nonblocking_download_ack and/or publication_nonblocking_download_ack scripts (recommended).

- certificate and certificate_password protocol options renamed
  The TLS and HTTPS certificate and certificate_password protocol options have been renamed to identity and identity_password, respectively. See “-x mlsrv16 option” [MobiLink - Server Administration].

MobiLink client changes

- Dbmlsync integration component deprecated
  The Dbmlsync integration component is deprecated. It has been replaced by the dbmlsync API.
  See “Dbmlsync API” [MobiLink - Client Administration].

- dbmlsync StreamCompression extended option no longer supported
  This option is no longer supported.

- -lt extended option now defaults to OFF
  The LockTables (-lt) extended option for dbmlsync formerly defaulted to ON. It now defaults to OFF, meaning that by default, dbmlsync does not lock synchronization tables. See “LockTables (lt) extended option” [MobiLink - Client Administration].

Miscellaneous MobiLink behavior changes

Server-initiated synchronization

- Sierra Wireless Aircards no longer supported
  The SMS listening libraries for Sierra Wireless Aircards are no longer supported.

- -g option deprecated
  The -g Listener option has been replaced by the -ni option.

- -ga option deprecated
  The -ga Listener option has been deprecated. Asynchronous IP tracking is now implicit.

- -gi default changed
  The default polling interval has changed from 10 seconds to 60 seconds.
SQL Remote

The following sections describe the new features, behavior changes, and deprecated features in SQL Remote for version 11.0.0.

SQL Remote new features

- **Enhancements to Extraction utility (dbxtract) -ea option**  The -ea option for dbxtract now accepts both none and simple as encryption types. Specifying none results in no encryption. Specifying simple results in simple encryption. Also, the default encryption type has changed, depending on whether the -ek, -et, or -ep options are specified with -ea. See “Extraction utility (dbxtract)” [SQL Remote].

- **-ap, -er, -et, and -nl options added to Extraction utility (dbxtract)**  You can now use the -ap, -er, -et, and -nl options with dbxtract. See “Extraction utility (dbxtract)” [SQL Remote].

SQL Remote behavior changes and deprecated features

Following is a list of changes to SQL Remote introduced in version 11.0.0.

- **LOAD TABLE now optionally logs loaded rows as inserts**  The new clause WITH ROW LOGGING means that SQL Anywhere remote databases can now load data using the LOAD TABLE statement. Previously, LOAD TABLE was not always useful in a synchronization or replicating environment because the rows were not logged and so they were ignored by dbremote or dbltm. See “Data import with the LOAD TABLE statement” [SQL Anywhere Server - SQL Usage].

- **VIM and MAPI message types unsupported**  Support for the VIM and MAPI message systems has been removed in this release. When you upgrade a database that uses VIM or MAPI to SQL Anywhere version 11.0.0, you must change the message type to File, FTP, or SMTP. Dbremote.exe does not start if the message type is MAPI or VIM.

  The simplest change is to switch to SMTP/POP; changing message types may require a change to your mail server to support SMTP/POP.

  For more information about choosing a SQL Remote message type, see “SQL Remote message systems” [SQL Remote].

UltraLite

The following sections describe the new features, behavior changes, and deprecated features in UltraLite for version 11.0.0.

UltraLite new features

Following is a list of additions to UltraLite introduced in version 11.0.0.
Main features

- **Support for synchronization profiles**  UltraLite 11.0.0 and 11.0.1 support synchronization profiles. See “ALTER SYNCHRONIZATION PROFILE statement [UltraLite]” [UltraLite - Database Management and Reference], “CREATE SYNCHRONIZATION PROFILE statement [UltraLite]” [UltraLite - Database Management and Reference], and “DROP SYNCHRONIZATION PROFILE statement [UltraLite]” [UltraLite - Database Management and Reference].

- **UltraLite SELECT statement**  The default for UltraLite SELECT statements that do not explicitly contain a FOR clause is now FOR READ ONLY. This change allows UltraLite to choose more optimal plans for queries when updates are not permitted. See “SELECT statement [UltraLite]” [UltraLite - Database Management and Reference].

- **UltraLite SYNCHRONIZE statement**  A new statement for synchronizing an UltraLite synchronization profile or specific synchronization options. See “SYNCHRONIZE statement [UltraLite]” [UltraLite - Database Management and Reference].

- **UltraLite CREATE SYNCHRONIZATION PROFILE statement**  A new statement for creating an UltraLite synchronization profile. Synchronization profiles define how an UltraLite database synchronizes with the MobiLink server. See “CREATE SYNCHRONIZATION PROFILE statement [UltraLite]” [UltraLite - Database Management and Reference].

- **UltraLite ALTER SYNCHRONIZATION PROFILE statement**  A new statement for altering an UltraLite synchronization profile. See “ALTER SYNCHRONIZATION PROFILE statement [UltraLite]” [UltraLite - Database Management and Reference].

- **UltraLite DROP SYNCHRONIZATION PROFILE statement**  A new statement for deleting an UltraLite synchronization profile. See “DROP SYNCHRONIZATION PROFILE statement [UltraLite]” [UltraLite - Database Management and Reference].

- **Support for SQL Anywhere passthrough scripts**  This functionality is not supported in UltraLite 12.

UltraLite utilities now include support for SQL Anywhere pass-through scripts. The changes apply to the following utilities:

- ulcond11
- ulunload
- ulload
- ulinfo
- ulsync

See:

- “UltraLite Database Unload utility (ulunload)” [UltraLite - Database Management and Reference]
- “UltraLite Load XML to Database utility (ulload)” [UltraLite - Database Management and Reference]
- “UltraLite Information utility (ulinfo)” [UltraLite - Database Management and Reference]
- “UltraLite Synchronization utility (ulsync)” [UltraLite - Database Management and Reference]
UltraLite database validation  You can now use the `ulvalid` utility or `ValidateDatabase` API to validate an UltraLite database. The validation tests for certain types of corruption in the database file, and you can use command line parameters to refine your results. See “UltraLite Validate Database utility (ulvalid)” [UltraLite - Database Management and Reference] and “Validating an UltraLite database” [UltraLite - Database Management and Reference].

UltraLite.NET now supports the `ValidateDatabase` function. You can now validate a database or specific tables with or without a connection. See “ULDatabaseManager class [UltraLite.NET]” [UltraLite - .NET Programming] and “ULConnection class [UltraLite.NET]” [UltraLite - .NET Programming].

You can now use the Validate Database Wizard in Sybase Central to validate an UltraLite database. The Validate Database option is available on the Tools menu.

Support for events and notifications  UltraLite now supports events and notifications. Notification messages are sent to registered queues or connections when events occur. User events may also be defined and triggered by applications. APIs for events and notifications are provided in each supported language. Additionally, a SQL function is provided to access the API functionality.

UltraLite support for isolation levels  Now, by default, connections are isolated from each other. Uncommitted changes by other connections and downloads are not visible until committed.

You can now set the isolation level to READ_COMMITTED or READ_UNCOMMITTED. See “Isolation levels” [UltraLite - Database Management and Reference] and “UltraLite transaction processing” [UltraLite - Database Management and Reference].

UltraLite.NET now supports the ReadUncommitted isolation level. The default isolation level of a connection in auto-commit mode is ReadCommitted. See “UltraLite transaction processing” [UltraLite - Database Management and Reference] and “Isolation levels” [UltraLite - Database Management and Reference].

UltraLite ALTER DATABASE SCHEMA FROM FILE statement  You can now use the `ALTER DATABASE SCHEMA FROM FILE` statement to alter an UltraLite schema. The `ALTER DATABASE SCHEMA FROM FILE` statement replaces the 9.0.2 schema upgrade feature that was implemented with the now removed `UpgradeSchemaFromFile` or `ApplyFile` methods. Use either the `ulinit` or `ulunload` utilities to ensure that the DDL statements required are syntactically correct.

See:

- “ALTER DATABASE SCHEMA FROM FILE statement [UltraLite]” [UltraLite - Database Management and Reference]
- “Deploying UltraLite database schema upgrades” [UltraLite - Database Management and Reference]
- “UltraLite Initialize Database utility (ulinit)” [UltraLite - Database Management and Reference]
- “UltraLite Database Unload utility (ulunload)” [UltraLite - Database Management and Reference]

Extract Database Wizard behavior changes  You can now exclude tables from the extraction process, and the Extract Database Wizard now omits publications with duplicate names from the list of available publications. See “UltraLite upgrades” on page 311.
• **UltraLite client version and build number added to MobiLink log files**  During synchronization, UltraLite clients now add their version and build number to the MobiLink server log. See:
  ○ “MobiLink server log viewing” [MobiLink - Server Administration]
  ○ “UltraLite synchronization parameters” [UltraLite - Database Management and Reference]

• **UltraLite LOAD TABLE statement**  The LOAD TABLE statement can now be executed on desktop computers. See UltraLite LOAD TABLE statement. See “LOAD TABLE statement [UltraLite]” [UltraLite - Database Management and Reference].

• **Background synchronization support**  You can now begin a synchronization on a separate thread at any point in your application and UltraLite will upload only the rows that were committed at the time the upload began. You can now modify the database during the upload and commit your changes without affecting the upload. Any rows committed while the upload is in progress are ignored by the upload. See “UltraLite concurrency” [UltraLite - Database Management and Reference].

• **Enhanced GUID identifier support**  In previous versions of UltraLite, runtime allowed the input and output of UUID (Universally Unique Identifier) or GUID (Globally Unique Identifier) identifiers as either 16-byte binaries or strings. An endian conversion made the identifiers compatible with GUID structs. In UltraLite 11, GUID structs can be explicitly input and output from the runtime and the endian conversion is not required.

• **ul_stream_error struct**  In UltraLite 11, the stream_id, stream_context, and error_string_length fields are removed. In addition, the error_string field has been changed from a user-supplied char * to a static char array.

---

**Platforms and devices**

• **Support for native amd64/x64 ESQL and C++ application deployment to 64 bit Windows platforms (64 bit XP and later)**  UltraLite now supports deploying native amd64/x64 ESQL and C++ applications to 64-bit Windows platforms (64-bit XP and later). Note, however, that to develop UltraLite applications on a 64-bit computer, you must use the 32-bit versions of the UltraLite utilities. In addition, you will need to use the 32-bit version of DBISQL and Sybase Central if you are connecting to UltraLite on a 64-bit computer.

• **UltraLite utilities ported to Linux (32 bit)**  See “Utilities” on page 253.

---

**Security**

• **End-to-end encryption**  UltraLite now supports protocol-level end-to-end encryption. Data is encrypted using 128-bit AES in cipher block chaining (CBC) mode with key exchange handled via RSA or ECC.
Utilities

- **UltraLite Unload Database to XML utility (ulunload)** You can now use the -s option to unload the schema and save data in a SQL Anywhere-compatible format. See “UltraLite Database Unload utility (ulunload)” [UltraLite - Database Management and Reference].

- **UltraLite Initialize Database utility (ulinit)** You can now use the -d option to copy data from a SQL Anywhere database into a new UltraLite database. See “UltraLite Initialize Database utility (ulinit)” [UltraLite - Database Management and Reference].

- **ulerase** New utility program to erase an UltraLite database (including any temporary files related to the database). This utility requires a user ID and password to confirm access to the database. See “UltraLite Erase utility (ulerase)” [UltraLite - Database Management and Reference].

- **ulvalid** New utility program to validate an UltraLite database. Validation tests for certain types of corruption in the database file, and is configurable according to command line parameters. See “UltraLite Validate Database utility (ulvalid)” [UltraLite - Database Management and Reference].

- The following UltraLite utilities have been ported to Linux (and are available in 32-bit versions only)
  - **ulcreate** create a new, empty UltraLite database
  - **ulerase** permanently erase an UltraLite database and associated checkpoint file
  - **ulinfo** display information about an existing UltraLite database
  - **ulinit** create a new UltraLite database from the schema available in a SQL Anywhere reference database
  - **ulload** create and load an UltraLite database from the XML data saved by ulunload
  - **ulsync** synchronize an UltraLite database with a consolidated database, using MobiLink as the transfer agent
  - **ununload** unload an UltraLite database to XML
  - **ulvalid** run validity checks on an UltraLite database

The ulunloadold utility is not available on Linux.

SQL

- **IF and CASE statements, and CASE expression enhancement** For improved compatibility, IF expressions are now permitted to end with either ENDIF or END IF. CASE expressions are now permitted to end with either END or END CASE. See “IF expressions” [UltraLite - Database Management and Reference] and “CASE expressions” [UltraLite - Database Management and Reference].
Programming interfaces

General improvements

Publication masks have been replaced by publication lists. The keyword Publications takes a comma-separated list of publication names.

UltraLite C/C++

A new method, UltraLite_Table* OpenTableEx(), is now part of the UltraLite_Connection object. This method gives non-SQL applications a more versatile way of opening a table and directly scan rows. See “Direct page scans” [UltraLite - Database Management and Reference].

With this method you can specify one of the following ways to open a table:

- To return the rows in the order of the primary key, use ul_table_open_primary_key.
- To return the rows in arbitrary order, use ul_table_open_no_index.
- To return rows in the order specified by an index, use ul_table_open_with_index.

UltraLite embedded SQL

- Documentation for two functions related to error interpretation. See “ULGetErrorParameter method [UltraLite Embedded SQL]” [UltraLite - C and C++ Programming] and “ULGetErrorParameterCount method [UltraLite Embedded SQL]” [UltraLite - C and C++ Programming].
- Functions ULGetLastDownloadTime, ULResetLastDownloadTime, and ULCountUploadRows have changed syntax to reflect the change from publication masks to publication lists.
- The function ULGetPublicationMask is no longer available.

UltraLite.NET

- ULDataReader class includes a new method: GetRowCount(threshold) to retrieve row count up to a specified maximum number of rows.
- The ULDataReader class now implements the IListSource interface.

UltraLite for M-Business Anywhere

- New methods in Connection class for event handling and notification: cancelGetNotification, createNotificationQueue, declareEvent, destroyNotificationQueue, getNotification, getNotificationParameter, registerForEvent, sendNotification, and triggerEvent.

UltraLiteJ

UltraLiteJ is a Java implementation of UltraLite that supports Java SE and Java ME environments, including BlackBerry smartphones. See “System requirements and supported platforms” [UltraLite - Java Programming].

UltraLite behavior changes and deprecated features

Following is a list of changes to UltraLite introduced in version 11.0.0.
Deprecated platforms

- The UltraLite C++ interface no longer supports the Symbian OS. Developers of UltraLite applications for Symbian should use UltraLiteJ.

- The UltraLite.NET interface no longer supports the .NET 1.0 component. The .NET 1.0 API reference has been removed from the documentation.

UltraLite for AppForge is not supported in version 11.

Database properties

The following database properties have been deprecated in this release:

- CollationName

Connection parameters

The following UltraLite connection parameter has been deprecated in this release.

- ORDERED_TABLE_SCAN

Removed utilities

The Migrate C++ Applications Wizard is no longer available in Sybase Central.

Removed, deprecated, and modified functions

- The ULGetPublicationMask function in the ESQL interface is removed (publication masks have been replaced by publication lists).

- Functions ULGetLastDownloadTime, ULResetLastDownloadTime and ULCountUploadRows in the ESQL interface have changed syntax to reflect the change from publication masks to publication lists.


- In the M-Business Anywhere and .NET APIs, the following methods used to accept a publication mask as a parameter. That parameter has been changed to a publication list (string). The affected methods are in the Connection class: countUploadRows, getLastDownloadTime, and resetLastDownloadTime.

- The ul_sync_info structure has changed: the fields disable_concurrency, checkpoint_store and table_order have been removed. These options are now specified in a new field named additionalParms that contains semicolon delimited keyword-value pairs.

Miscellaneous

- Running a second instance of CustDB In previous versions, running a second instance of a CustDB application, would cause an error. Now, if you start a second instance of CustDB, the first instance is now brought to the foreground and the second instance exits.
Sybase Central and Interactive SQL

The following sections describe the new features, behavior changes, and deprecated features in Sybase Central and Interactive SQL for version 11.0.0.

Sybase Central and Interactive SQL new features

Following is a list of additions to Sybase Central and Interactive SQL introduced in version 11.0.0.

- **New fast launcher strategy**  
  Previously, the Interactive SQL and Sybase Central fast launchers were started when a user logged in. Now, fast launchers are started only when Interactive SQL or Sybase Central is started. Then, the fast launcher continues to run for 30 minutes (configurable) after the application is shut down. This new strategy speeds up subsequent launches of Interactive SQL and Sybase Central done within the 30 minute window. See “Fast launcher option” [SQL Anywhere Server - Database Administration].

- **Connect window enhancements and changes**  
  The following enhancements have been made to the Connect window in Sybase Central and Interactive SQL:
  - **New Save As ODBC Data Source tool**  
    The Save As ODBC Data Source tool allows you to generate an ODBC data source using the connection parameters that you specify in the Connect window. To use this tool, click the Tools button on the Connect window, click Save As ODBC Data Source, and follow the onscreen instructions. See “Creating an ODBC data source (administration tools)” [SQL Anywhere Server - Database Administration].
  - **New Copy Connection String To Clipboard tool**  
    The Copy Connection String To Clipboard tool allows you to copy the connection string to the clipboard to see what is being passed to the database server. To use this tool, click the Tools button on the Connect window, and click Copy Connection String To Clipboard. Paste the connection string in a text editor to view it.
  - **Connect Assistant**  
    There is now a Connect Assistant tool in the right half of the Connect window. The Connect Assistant is a wizard-like feature designed to help you connect to a database.
  - **Enhancements to the Advanced tab**  
    The Advanced tab now displays a table of properties, as well as brief descriptions of the properties themselves.
  - **New Networking tab**  
    The Networking tab allows you to specify options for the supported protocols: shared memory and TCP/IP. This new tab replaces the functionality offered by the Search Network For Database Servers option that was previously located on the Database tab.

- **Keyboard shortcuts**  
  Now in the Sybase Central Code Editor and in the SQL Statements pane in Interactive SQL, you can use keyboard shortcuts to increase and decrease the indentation of your code and comments. In addition you can use keyboard shortcuts to add and remove both the double hyphen and double slash comment indicators. See “Sybase Central keyboard shortcuts” [SQL Anywhere Server - Database Administration].
Sybase Central new features

Following is a list of additions to Sybase Central plug-ins introduced in version 11.0.0.

SQL Anywhere plug-in new features

- **Database Overview tab**  You can now obtain a high-level view of the health and statistics of the database server and its features. See “Database health and statistics” [SQL Anywhere Server - Database Administration].

- **Database documentation**  You can now generate documentation for your database using the Database Documentation Wizard. See “Documenting a database” [SQL Anywhere Server - Database Administration].

- **New wizards**  The SQL Anywhere plug-in now contains the following new wizards:
  - Create Login Policy Wizard
  - Database Documentation Wizard
  - Create Text Index Wizard
  - Create Text Configuration Object Wizard
  - Create Database Wizard

- **New properties windows**  The SQL Anywhere plug-in now contains properties windows for the following features:
  - Text indexes
  - Text object configurations
  - External environments
  - Materialized views
  - Database auditing

- **Set auditing preferences within Sybase Central**  You can now set preferences for database auditing and view auditing information using Sybase Central. See “Configuring auditing (Sybase Central)” [SQL Anywhere Server - Database Administration] and “Retrieving auditing information (Sybase Central)” [SQL Anywhere Server - Database Administration].

- **Support for specifying collation tailoring settings**  The Settings tab on the Database Properties window now allows you to specify collation tailoring options. You can also specify collation tailoring options when creating a new database using the Create Database Wizard.

UltraLite plug-in new features

- **New wizards**  The UltraLite plug-in now contains the following new wizards:
  - Validate Database Wizard
  - Create Synchronization Profile Wizard

- **New properties windows**  The UltraLite plug-in now contains properties windows for the following features:
  - Synchronization profiles
MobiLink plug-in new features

- **Support for synchronization profiles** You can create and manage synchronization profiles in Sybase Central.

- **New wizards** The MobiLink plug-in now contains the following wizards:
  - Create Passthrough Script Wizard
  - Create Passthrough Download Wizard

- **New properties windows** The MobiLink plug-in now contains property windows for the following features:
  - Passthrough scripts
  - Passthrough downloads

Interactive SQL new features

Following is a list of additions to Interactive SQL introduced in version 11.0.0.

- **Automatically release database locks** The database server creates schema locks on tables that you view in Interactive SQL, even if you do not modify the table.

  However, now Interactive SQL attempts to release the database schema locks it creates when it displays your result set.

  After you execute a statement that returns a result set, Interactive SQL checks if your connection has any uncommitted changes in the database. If none exist, then Interactive SQL releases your schema locks; otherwise, Interactive SQL does not release your schema locks. That is, Interactive SQL does not release your schema locks if you have any uncommitted changes to the database.

- **Support for read-only result sets** You can now make result sets read-only in Interactive SQL. To do this, click Options > Results, and then click the Disable Editing option. This setting applies to subsequently fetched results. See “Editing a row in an Interactive SQL result set” [SQL Anywhere Server - Database Administration].

  When deploying Interactive SQL, you can prevent users from changing this setting by adding disableResultSetEditing to the entry for lockedPreferences in the OEM.ini file. See “Administration tools configuration” [SQL Anywhere Server - Programming].

- **Execute SQL statements one statement at a time** Previously, if you wanted to execute SQL statements one at a time, you repeatedly selected the SQL statement and clicked Execute Selection. Now you can click Single Step to execute the selected statement and to select the next statement for execution. Similar to Execute Selection, Single Step is available from the SQL menu in Interactive SQL. See “SQL statements in Interactive SQL” [SQL Anywhere Server - Database Administration].

- **SQL file and database connection favorites** You can now create and maintain a list of favorite database connections and a list of favorite SQL script files with the Favorites menu in Interactive SQL. See “Adding SQL script files, SQL statements, and connections to favorites” [SQL Anywhere Server - Database Administration].
● **Disable table editing of result sets**  When deploying Interactive SQL, you can now disable table editing of SQL Anywhere and UltraLite result sets in Interactive SQL. See “Administration tools configuration” [SQL Anywhere Server - Programming].

● **New keyboard shortcuts**  New keyboard shortcuts have been added to Interactive SQL. See “Keyboard shortcuts for Interactive SQL” [SQL Anywhere Server - Database Administration].

● **Support for preventing option changes in client applications**  You can now prevent users from changing some of the Interactive SQL option settings by locking the settings in the OEM.ini file. See “Administration tools configuration” [SQL Anywhere Server - Programming].

● **New -version option for dbisql**  At a command prompt, type `dbisql -version` to see the version number of Interactive SQL. See “Interactive SQL utility (dbisql)” [SQL Anywhere Server - Database Administration] and “Interactive SQL for UltraLite utility (dbisql)” [UltraLite - Database Management and Reference].

---

**Sybase Central and Interactive SQL behavior changes and deprecated features**

Following is a list of changes to Sybase Central and Interactive SQL introduced in version 11.0.0.

● **Database tool launcher executables are easier to redeploy**  The launcher executables for the Sybase Central, Interactive SQL, the Database Console utility, and the MobiLink Monitor are now easier to redeploy. Registry entries and a set directory structure for the location of the JAR files are no longer required. Each executable needs to have a corresponding `filename.INI` file in the same directory (with the same name) as the `filename.exe` file. The .INI file contains the details on how to load the tool. See “Administration tool deployment” [SQL Anywhere Server - Programming].

● **OEM.ini [help] section no longer supported**  The [help] section in the OEM.ini file is no longer supported. For more information, see “Administration tools configuration” [SQL Anywhere Server - Programming].

---

**Sybase Central behavior changes and deprecated features**

Following is a list of changes to Sybase Central introduced in version 11.0.0.

● **Sybase Central configuration file renamed**  The `.screpository` file is now named `.screpository600`.

---

**SQL Anywhere plug-in changed features**

● **Enhancements to properties windows**  The following property pages have been updated:

  ○ **Web Service Properties** window
  ○ **User Properties** window
  ○ **View Properties** windows
  ○ **Materialized View Properties** window
**Debugging specific users**  When you enter *Debug* mode in the SQL Anywhere plug-in, you must specify what users you want to debug. See “Tutorial: Getting started with the debugger” [*SQL Anywhere Server - SQL Usage*].

**Updated wizards**  The following wizards have been updated:
- Create User Wizard
- Create Web Services Wizard
- Deployment Wizard

**MobiLink plug-in changed features**

**Updated wizards**  The following wizards have been updated:
- Deploy Synchronization Model Wizard

**SQL Anywhere plug-in deprecated features**

**Removed properties tabs**  The following property windows have been updated:
- Table Properties window
- UltraLite Project Properties window
- UltraLite Statement Properties window

### Interactive SQL behavior changes and deprecated features

Following is a list of changes to Interactive SQL introduced in version 11.0.0.

**Graphical plans are now viewable in a Plan Viewer**  You now view graphical plans for SQL Anywhere databases in a separate, resizable window called the Plan Viewer, in Interactive SQL. This change makes it easier to view and compare plans because you can now open multiple Plan Viewer windows at the same time. To access the Plan Viewer, click **Tools » Plan Viewer**. Text plans for UltraLite databases are also displayed in the Plan Viewer. See “Creating a graphical plan with detailed and node statistics” [*SQL Anywhere Server - Database Administration*].

In addition, the Interactive SQL option `isql_plan` option is unsupported.

**Support for viewing long and short plans has been removed**  You can no longer view text plans for SQL Anywhere databases in Interactive SQL. However, you can still retrieve them using the `EXPLANATION` and `PLAN` functions. You can still view text plans for UltraLite databases using the Plan Viewer in Interactive SQL.

**Printing execution plans and result sets**  Now you can print the contents of the SQL Statements pane and the result sets by pressing Ctrl+P or by clicking **Print** from the File menu. Previously you could only print the contents of the SQL Statements pane. You can print in the Plan Viewer by clicking the **Print** button.

**Line numbers added to the SQL Statements pane**  Line numbers are now visible on the left side of the SQL Statements pane. These line numbers can help you identify the location of syntax errors.
● **Enhancement to the Execute SQL Statements toolbar button**  Previously, on the Interactive SQL toolbar, the *Execute SQL Statements* button executed all SQL statements. Now you can specify whether to execute all statements, or to execute only the selected statements when the button is clicked.

To set the behavior of the *Execute SQL Statements* button, in the *Tools* menu, click *Options* » *Toolbar*. See “SQL statements in Interactive SQL” [*SQL Anywhere Server - Database Administration*].

● **Enhancement to executing batch statements**

○ Interactive SQL provides improved feedback when executing batches of statements. When executing SQL statements from the *SQL Statements* pane, the statement being executed is now selected and scrolled into view. When executing script files by clicking *File* » *Run Script*, a status window appears that shows the progress through the script. See “Executing SQL statements (Interactive SQL)” [*SQL Anywhere Server - Database Administration*].

● **Enhancements to the Results pane**

○ In the *Results* pane, you can now select all the results by pressing Ctrl+A. You can also select the entire result set, not just the currently fetched results. When the *Results* pane does not contain the entire result set, you are prompted to fetch the remaining results. Otherwise, only the currently fetched results are selected.

○ Now when you copy cells from the *Results* tab the copied data is formatted based on the following Interactive SQL options: isql_field_separator, isql_quote, and isql_escape_character. You can also copy to the clipboard selected values, rows, and columns from the result set. See “Copying rows, columns, and cells from an Interactive SQL result set” [*SQL Anywhere Server - Database Administration*].

○ Now when you click a column-header in the *Results* tab, the results are sorted by that column. When the *Results* pane does not contain the entire result set, you are prompted to fetch the remaining results. Otherwise, only the currently fetched results are sorted.

○ Now you can generate and copy to the clipboard INSERT, DELETE, and UPDATE statements that are based on selected rows in the result set. See “Generating a SQL statement from Interactive SQL result sets” [*SQL Anywhere Server - Database Administration*].

○ The *Results* pane in Interactive SQL has been enhanced to include the following features, available from the right-click menu:

  ● **Copy » Copy Cell**  Copies the contents of the selected cell.

  ● **Copy » Copy Column**  Copies cell values from the column the selected cell.

  ● **Generate » INSERT Statement**  Generates an INSERT statement for each selected row and copies them to the clipboard.

  ● **Generate » DELETE Statement**  Generates a DELETE statement for each selected row and copies them to the clipboard.
Generate UPDATE Statement
Generates an UPDATE statement for each selected row and copies them to the clipboard. The generated statements set the column values to their current values. Consequently, executing the statements would not actually change the column values. This functionality can be useful for providing a template UPDATE statement which you could edit before executing it.

See “Copying rows, columns, and cells from an Interactive SQL result set” [SQL Anywhere Server - Database Administration] and “Generating a SQL statement from Interactive SQL result sets” [SQL Anywhere Server - Database Administration].

Enhancements to Interactive SQL statements

- DESCRIBE statement enhancements
  The DESCRIBE statement can now return information about the database or database server that is connected to Interactive SQL. See “DESCRIBE statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference].

- INPUT and READ statement enhancements
  The INPUT and READ statements now attempt to resolve relative paths in two ways. See “INPUT statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference] and “READ statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference].

- Enhancements to the INPUT and OUTPUT statements
  
  - New support for importing from, and exporting to, ODBC sources
    You can now specify an ODBC data source when importing into and exporting from the database using the INPUT and OUTPUT statements. To do so, use the new USING clause. See “INPUT statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference] and “OUTPUT statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference].

    You can also specify an ODBC data source when using the Import Wizard and the Export Wizard. See “Importing data with the Import Wizard” [SQL Anywhere Server - SQL Usage] and “Exporting data with the Export Wizard” [SQL Anywhere Server - SQL Usage].

  - New support for byte order mark (BOM)
    You can now control whether a byte order mark (BOM) in data is processed. To do so, use the new BYTE ORDER MARK clause. See “INPUT statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference] and “OUTPUT statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference].

  - Supported formats for the INPUT statement have changed
    The INPUT statement no longer supports the dBase, Lotus, Excel, and FoxPro file formats. TEXT and FIXED are still supported. If you want to continue to use these file formats, you must do so via an ODBC driver. See “INPUT statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference].

  - Supported formats for the OUTPUT statement have changed
    The OUTPUT statement no longer supports the dBase, Lotus, Excel, and FoxPro file formats. TEXT, FIXED, HTML, SQL, and XML are still supported. See “OUTPUT statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference].

  - ASCII format is renamed TEXT for INPUT and OUTPUT statements for the dbisqlc utility
    The following formats have been renamed in this release when using the INPUT and OUTPUT statements with the dbisqlc utility:
Changes to the Import Wizard and the Export Wizard  
When the Import Wizard or the Export Wizard finishes, the SQL statement generated by the wizard is stored in the command history. To view the generated SQL Statement, click SQL » History.

Interactive SQL options

- `isql_allow_read_client_file` and `isql_allow_write_client_file`  
These two options describe how Interactive SQL responds to requests to read and write client-side files. See “`isql_allow_read_client_file` option [Interactive SQL]” [SQL Anywhere Server - Database Administration] and “`isql_allow_write_client_file` option [Interactive SQL]” [SQL Anywhere Server - Database Administration].

- `-codepage` option deprecated  
If you want Interactive SQL to read a file with a specific code page, use the ENCODING clause of the INPUT, OUTPUT, or READ statement. See:
  - “Interactive SQL utility (dbisql)” [SQL Anywhere Server - Database Administration]
  - “Interactive SQL for UltraLite utility (dbisql)” [UltraLite - Database Management and Reference]
  - “INPUT statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference]
  - “OUTPUT statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference]
  - “READ statement [Interactive SQL]” [SQL Anywhere Server - SQL Reference]

- `isql_plan` option unsupported  
The Interactive SQL option `isql_plan` option is no longer supported. Attempts to set it are silently ignored for backward compatibility. See “Creating a graphical plan with detailed and node statistics” [SQL Anywhere Server - Database Administration].

- SET OPTION statement PUBLIC keyword removed  
Support for the PUBLIC keyword is removed for setting Interactive SQL options using the SET OPTION statement. See “Interactive SQL options” [SQL Anywhere Server - Database Administration].

Changes to the Interactive SQL launcher  
The executable for the Windows version of the Interactive SQL launcher has changed from `dbisqlg.exe` to `dbisql.exe`.

The executable for the command-line version of the Interactive SQL launcher changed from `dbisql.exe` to `dbisql.com`. Batch scripts should call `dbisql` or `dbisql.com`, not `dbisql.exe`.

SQL Anywhere Console utility behavior changes

Following is a list of changes the SQL Anywhere Console utility introduced in version 11.0.0.

- New Console options  
You can now specify a date and time for the database server to shut down from the Options window. In the File menu, click Options » Console.
MobiLink Monitor behavior changes

Following is a list of changes to the MobiLink Monitor introduced in version 11.0.0.

The MobiLink Monitor cannot read monitor files that were created with a version 9 or earlier MobiLink server; the MobiLink Monitor should only be used with MobiLink servers of the same major version. In addition, the worker column has been removed and the following MobiLink Monitor properties have been renamed:

<table>
<thead>
<tr>
<th>Old property name</th>
<th>New property name</th>
</tr>
</thead>
<tbody>
<tr>
<td>preload_upload</td>
<td>sync_request</td>
</tr>
<tr>
<td>verify_upload</td>
<td>authenticate_user</td>
</tr>
</tbody>
</table>

Documentation enhancements

- **Documentation directory enhancements**  
  Previously the documentation existed in `%SQLANY11%\docs`; now the documentation resides in `%SQLANY11%\documentation`. The file names for the individual HTML Help and PDF files are also updated.

  The new file names for the HTML Help are the following:

  - dbadmin_en11.chm
  - dbprogramming_en11.chm
  - dbreference_en11.chm
  - dbusage_en11.chm
  - mlclient_en11.chm
  - mlserver_en11.chm
  - mlisysync_en11.chm
  - mlstart_en11.chm
  - sachanges_en11.chm
  - saerrors_en11.chm
  - saintro_en11.chm
  - scplugin_en11.chm
  - sqlanywhere_en11.chm
  - sqlremote_en11.chm
  - uladmin_en11.chm
  - ulc_en11.chm
  - uldotnet_en11.chm
  - ulj_en11.chm
  - ulmbus_en11.chm

- **Supported platform pages are now accessed through the web site**  
  Previously, the supported platform pages were installed with the software. Now all supported platform information is available on the Sybase web site: [http://www.sybase.com/detail?id=1002288](http://www.sybase.com/detail?id=1002288).
Product-wide features

The following sections describe the new features, behavior changes, and deprecated features that affect all components of SQL Anywhere version 11.0.0.

Product-wide new features

Following is a list of product-wide additions introduced in version 11.0.0.

- **Error reporting enhancements**  On Windows, Windows Mobile, and Linux, when an error report is generated, a window appears where you can view the contents of the error report before choosing whether to submit it to iAnywhere. See “Troubleshooting: Reporting an error in SQL Anywhere” [SQL Anywhere Server - Database Administration].

  You can now configure the Support utility (dbsupport) using the -ce option, which sends an email when dbsupport is monitoring an application and that application crashes. See “Support utility (dbsupport)” [SQL Anywhere Server - Database Administration].

- **Additional Windows Mobile platform support**  SQL Anywhere now supports Windows Mobile 5 for smartphone and Windows Mobile 6 Standard edition. For information about running SQL Anywhere Server on Windows Mobile, see “SQL Anywhere for Windows Mobile” [SQL Anywhere Server - Database Administration] and “Installation considerations: Limitations on Windows Mobile 5 and 6 for smartphone” [SQL Anywhere Server - Database Administration].

- **New table in the SQL Anywhere sample database (demo.db)**  A new table, MarketingInformation, has been added to the SQL Anywhere sample database. Each row in this table holds an HTML page describing a product in the Products table. This table was added to provide richer character data to query on when testing and trying out features. See “SQL Anywhere sample database” [SQL Anywhere 16 - Introduction].

Product-wide behavior changes

Following is a list of product-wide changes in version 11.0.0.

- **Windows CE has changed to Windows Mobile**  The name Windows CE has been changed to be Windows Mobile in the documentation and the software, except where it is more accurate to continue referring to Windows CE.

- **Readcert, gencert, and reqtool removed**  The utilities readcert, gencert, and reqtool have been removed. They were previously deprecated. In their place, you can use createcert and viewcert. See “Certificate utilities” [SQL Anywhere Server - Database Administration].
- **Createcert and viewcert utilities supported on Mac OS X** The createcert and viewcert certificate utilities are now supported on Mac OS X. See “Certificate utilities” [SQL Anywhere Server - Database Administration].

- **Certificate and certificate_password protocol options renamed** The TLS and HTTPS certificate and certificate_password protocol options have been renamed to identity and identity_password, respectively. See:
  - SQL Anywhere database servers. See “-ec database server option” [SQL Anywhere Server - Database Administration].
  - SQL Anywhere web servers. See “-xs database server option” [SQL Anywhere Server - Database Administration].
  - SQL Anywhere protocol options. See “Identity protocol option” [SQL Anywhere Server - Database Administration] and “Identity_Password protocol option” [SQL Anywhere Server - Database Administration].
  - MobiLink servers. See “-x mlsrv16 option” [MobiLink - Server Administration].

- **Sample identity file changes** The identity files containing the sample certificates and corresponding private key for TLS have been renamed in this release. The file rsaserver.crt has been renamed rsaserver.id, and the file sample.crt has been renamed eccserver.id. The password for both of these identity files has been changed from tJ1#m6+W to test.

- **Changes to installation directories** 32-bit software is now installed to the bin32 directory instead of the win32 directory, and 64-bit software is installed to the bin64 directory, instead of the X64 directory. For example, in previous versions, software that was installed to C:\Program Files \SQL Anywhere 11\win32 is now installed to C:\Program Files\SQL Anywhere 11\bin32.

- **Changes to ODBC data sources for the sample databases** In previous releases, the ODBC data sources for the sample databases that are installed with the software were user data sources. The SQL Anywhere 11 Demo, and SQL Anywhere 11 CustDB data sources are now system data sources.

- **.NET 1.0 unsupported** SQL Anywhere 11 does not support Microsoft Visual Studio .NET 2002 or Visual Studio .NET 2003. However, Microsoft Visual Studio 2005 (.NET 2.0) and Visual Studio 2008 (.NET 3.x) are supported.
How to upgrade to SQL Anywhere 16

Differences between major releases, minor releases and Support Packages

- **Support Package**  A Support Package is a subset of the software with one or more bug fixes. The bug fixes are listed in the release notes for the update. Bug fix updates may only be applied to installed software with the same version number. While some testing has been performed on the software, do not distribute these files with your application unless you have thoroughly tested your application with the software.

- **Minor releases**  A minor release is a complete set of software that upgrades installed software from an older version with the same major version number (version number format is `major.minor.build`). Bug fixes and other changes are listed in the release notes for the upgrade.

- **Major releases**  A major release is a complete set of software that has its own version number.

Compatibility with existing databases, servers, and administration tools

See the Components by Platforms page for information about:

- Support for database and server connections to databases, servers, and client applications that use software from previous versions.

- Administration tool support to connect to databases, servers, and client applications that use software from previous versions.

See also

- “Unload utility (dbunload)” [SQL Anywhere Server - Database Administration]
- “How to ensure that you are running the correct version of the utilities when you have multiple versions installed” on page 275

Features requiring a rebuilt (unload/reload) database, upgraded database, or updated client libraries

While many features are available when you run an older database on the latest version of the database server, to access some features you must rebuild (unload and reload) your database, perform an upgrade on the database file, or update your client libraries.

**Note**

Features not listed in the categories below only require a SQL Anywhere 16.0 database server; they do not require existing databases to be upgraded or the client libraries to be updated. For a complete list of new features in SQL Anywhere 16, see “What's new in version 16.0” on page 1.
Differences between a rebuilt (unload/reload) database and an upgraded database

- **Upgraded databases**  An upgraded database is a database that has had some of its features updated. For example, an upgrade updates the system tables and views, adds new database options, recreates all system stored procedures, installs jConnect support, and changes support for Java in the database. Upgrade a database by using the Upgrade utility (dbupgrad) or the Upgrade Database Wizard in Sybase Central.

- **Rebuilt databases**  A rebuilt database is a new database that contains the data from an older database. Rebuild a database by using the UNLOAD statement or the Unload utility (dbunload) with the -an option.

Features that require a rebuild (unload/reload) of your database

- UltraLite version 16 cannot connect to databases that were created with earlier versions of UltraLite. Therefore, an unload/reload is required. However, UltraLite version 11 or later clients can synchronize with MobiLink version 16.

- The ADD COLUMN clause of the ALTER TABLE statement (version 10 and earlier databases). See “ALTER TABLE statement” [SQL Anywhere Server - SQL Reference].

- CESU-8 character set support. For information about supported collations, see “Alternate collations” [SQL Anywhere Server - Database Administration].

Features that require only an upgrade (dbupgrad utility or UPGRADE DATABASE statement) Features that require only an upgraded database also work if you unload/reload your database.

- SMTP and MAPI new error codes and changed return codes. See “Return codes for MAPI and SMTP system procedures” [SQL Anywhere Server - SQL Reference].

- ADD COLUMN clause of the ALTER TABLE statement (version 11 and later databases). See “ALTER TABLE statement” [SQL Anywhere Server - SQL Reference].

- CREATE INDEX statement. See “CREATE INDEX statement” [SQL Anywhere Server - SQL Reference].

- Roles and privileges, including changes made to Sybase Central to support the new role-based security model. See “New security model: Role-based access control (RBAC)” on page 7.

- Some system procedures are now run with the privileges of the invoker, rather than the definer (owner). See “Set procedures and functions to run with owner or invoker privileges” [SQL Anywhere Server - SQL Usage].

- LDAP user authentication support. See “LDAP user authentication” [SQL Anywhere Server - SQL Usage].

- ROW and ARRAY data type support. See “Composite data types” [SQL Anywhere Server - SQL Reference].

- OData support. See “OData support” [SQL Anywhere Server - Programming].
Features requiring a rebuilt (unload/reload) database, upgraded database, or updated client libraries

- Reserved words:
  - array
  - json
  - row
  - rowtype
  - unnest
  - varray

- Login policy options:
  - ldap_primary_server
  - ldap_secondary_server
  - ldap_auto_failback_period
  - ldap_failover_to_std
  - ldap_refresh_dn
  - change_password_dual_control

See “Root login policy” [SQL Anywhere Server - Database Administration].

- Database options:
  - “auto_commit_on_create_local_temp_index option” [SQL Anywhere Server - Database Administration]
  - “disk_sandbox option” [SQL Anywhere Server - Database Administration]
  - “-sbx database option” [SQL Anywhere Server - Database Administration]
  - “min_role_admins option” [SQL Anywhere Server - Database Administration]
  - “db_publisher option” [SQL Anywhere Server - Database Administration]
  - “login_mode option” [SQL Anywhere Server - Database Administration]
  - “trusted_certificates_file option” [SQL Anywhere Server - Database Administration]
  - “-al database option” [SQL Anywhere Server - Database Administration]

- Database server options:
  - “-gu database server option” [SQL Anywhere Server - Database Administration]
  - “-gk database server option” [SQL Anywhere Server - Database Administration]
  - “-gd database server option” [SQL Anywhere Server - Database Administration]
  - “-gl database server option” [SQL Anywhere Server - Database Administration]
  - “-al database server option” [SQL Anywhere Server - Database Administration]

- Connection properties:
  - “auto_commit_on_create_local_temp_index connection property” [SQL Anywhere Server - Database Administration]
  - “extern_login_credentials connection property” [SQL Anywhere Server - Database Administration]
System views:

- “SYSGROUP compatibility view” [SQL Anywhere Server - SQL Reference]
- “SYSGROUPS compatibility view” [SQL Anywhere Server - SQL Reference]
- “SYSLDAPSERVER system view” [SQL Anywhere Server - SQL Reference]
- “SYSRULEGRANT system view” [SQL Anywhere Server - SQL Reference]
- “SYSRULEGRANTS consolidated view” [SQL Anywhere Server - SQL Reference]
- “SYSRULEGRANTTEXT system view” [SQL Anywhere Server - SQL Reference]
- “SYSUSERAUTHORITY compatibility view (deprecated)” [SQL Anywhere Server - SQL Reference]
- “SYSTABLEPERM system view” [SQL Anywhere Server - SQL Reference]
- “SYSTABAUTH consolidated view” [SQL Anywhere Server - SQL Reference]
- “SYSCERTIFICATE system view” [SQL Anywhere Server - SQL Reference]
- “SYSTABLECOL system view” [SQL Anywhere Server - SQL Reference]
- “SYSTABLE system view” [SQL Anywhere Server - SQL Reference]
- “SYSCOLSTAT system view” [SQL Anywhere Server - SQL Reference]
- “SYSCOLSTATS consolidated view” [SQL Anywhere Server - SQL Reference]

System tables:

- “ISYSCERTIFICATE system table” [SQL Anywhere Server - SQL Reference]
- “ISYSTABLECOL system table” [SQL Anywhere Server - SQL Reference]
- “ISYSTAB system table” [SQL Anywhere Server - SQL Reference]
- “ISYSUSER system table” [SQL Anywhere Server - SQL Reference]

Timestamps are now stored as their UTC (Coordinated Universal Time) based equivalent in system tables, but system views expose both UTC timestamps and local timestamps. Applications that rely on the number of columns in the following system views and/or their underlying tables must be updated:

- “SYSTAB system view” [SQL Anywhere Server - SQL Reference]
- “SYSUSER system view” [SQL Anywhere Server - SQL Reference]
- “SYSTYPE view” [SQL Anywhere Server - SQL Reference]
- “SYSOBJECT system view” [SQL Anywhere Server - SQL Reference]
- “SYSTEXTIDX system view” [SQL Anywhere Server - SQL Reference]
- “SYSEXTERNVENVOBJECT system view” [SQL Anywhere Server - SQL Reference]
- “SYSHISTORY system view” [SQL Anywhere Server - SQL Reference]
- “SYSREMOTEUSER system view” [SQL Anywhere Server - SQL Reference]
- “SYSREMOTEUSERS consolidated view” [SQL Anywhere Server - SQL Reference]
- “SYSCOLSTAT system view” [SQL Anywhere Server - SQL Reference]
- “SYSCOLSTATS consolidated view” [SQL Anywhere Server - SQL Reference]

Functions:

- “ARRAY constructor [Composite]” [SQL Anywhere Server - SQL Reference]
- “ROW constructor [Composite]” [SQL Anywhere Server - SQL Reference]
- “ENCRYPT function [String]” [SQL Anywhere Server - SQL Reference]
- “DECRYPT function [String]” [SQL Anywhere Server - SQL Reference]
Features requiring a rebuilt (unload/reload) database, upgraded database, or updated client libraries

- Operators:
  - “UNNEST array operator” [SQL Anywhere Server - SQL Reference]

- System procedures:
  - “sa_certificate_info system procedure” [SQL Anywhere Server - SQL Reference]
  - “sa_cpu_topology system procedure” [SQL Anywhere Server - SQL Reference]
  - “sa_db_option system procedure” [SQL Anywhere Server - SQL Reference]
  - “sa_get_ldapserver_status system procedure” [SQL Anywhere Server - SQL Reference]
  - “sa_get_user_status system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_parse_json system procedure” [SQL Anywhere Server - SQL Reference]
  - “sa_server_option system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_copy_directory system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_copy_file system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_create_directory system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_delete_directory system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_delete_file system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_move_directory system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_move_file system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_create_secure_feature_key system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_alter_secure_feature_key system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_drop_secure_feature_key system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_list_secure_feature_keys system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_use_secure_feature_key system procedure” [SQL Anywhere Server - SQL Reference]
  - “xp_get_mail_error_code system procedure” [SQL Anywhere Server - SQL Reference]
  - “xp_get_mail_error_text system procedure” [SQL Anywhere Server - SQL Reference]
  - “xp_getenv system procedure” [SQL Anywhere Server - SQL Reference]
  - “xp_startmail system procedure” [SQL Anywhere Server - SQL Reference]
  - “xp_startsmtp system procedure” [SQL Anywhere Server - SQL Reference]
  - “xp_sendmail system procedure” [SQL Anywhere Server - SQL Reference]
  - “xp_stopmail system procedure” [SQL Anywhere Server - SQL Reference]
  - “xp_stopsmtp system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_objectpermission system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_displayroles system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_has_role system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_proc_priv system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_auth_sys_role_info system procedure” [SQL Anywhere Server - SQL Reference]
  - “sp_sys_priv_role_info system procedure” [SQL Anywhere Server - SQL Reference]
  - “sa_get_ldapserver_status system procedure” [SQL Anywhere Server - SQL Reference]
  - “sa_get_user_status system procedure” [SQL Anywhere Server - SQL Reference]
New or enhanced SQL statements:

- “GRANT statement” [SQL Anywhere Server - SQL Reference]
- “REVOKE statement” [SQL Anywhere Server - SQL Reference]
- “CREATE ROLE statement” [SQL Anywhere Server - SQL Reference]
- “ALTER ROLE statement” [SQL Anywhere Server - SQL Reference]
- “DROP ROLE statement” [SQL Anywhere Server - SQL Reference]
- “GRANT ROLE SYS_REPLICATION_ADMIN_ROLE statement [MobiLink] [SQL Remote]” [SQL Anywhere Server - SQL Reference]
- “GRANT ROLE SYS_RUN_REPLICATION_ROLE statement [MobiLink] [SQL Remote]” [SQL Anywhere Server - SQL Reference]
- “GRANT PUBLISH statement [SQL Remote]” [SQL Anywhere Server - SQL Reference]
- “REVOKE PUBLISH statement [SQL Remote]” [SQL Anywhere Server - SQL Reference]
- “CREATE LDAP SERVER statement” [SQL Anywhere Server - SQL Reference]
- “ALTER LDAP SERVER statement” [SQL Anywhere Server - SQL Reference]
- “DROP LDAP SERVER statement” [SQL Anywhere Server - SQL Reference]
- “VALIDATE LDAP SERVER statement” [SQL Anywhere Server - SQL Reference]
- “ALTER USER statement” [SQL Anywhere Server - SQL Reference]
- “CREATE LOGIN POLICY statement” [SQL Anywhere Server - SQL Reference]
- “ALTER LOGIN POLICY statement” [SQL Anywhere Server - SQL Reference]
- “COMMENT statement” [SQL Anywhere Server - SQL Reference]
- “SET MIRROR OPTION statement” [SQL Anywhere Server - SQL Reference]
- “CREATE CERTIFICATE statement” [SQL Anywhere Server - SQL Reference]
- “DROP CERTIFICATE statement” [SQL Anywhere Server - SQL Reference]
- “CREATE FUNCTION statement [Web service]” [SQL Anywhere Server - SQL Reference]
- “CREATE PROCEDURE statement [Web service]” [SQL Anywhere Server - SQL Reference]
- “CREATE TEXT INDEX statement” [SQL Anywhere Server - SQL Reference]
- “CREATE TEMPORARY TRACE EVENT SESSION statement” [SQL Anywhere Server - SQL Reference]
- “ALTER TRACE EVENT SESSION statement” [SQL Anywhere Server - SQL Reference]
- “DROP TRACE EVENT statement” [SQL Anywhere Server - SQL Reference]
- “DROP TRACE EVENT SESSION statement” [SQL Anywhere Server - SQL Reference]
- “NOTIFY TRACE EVENT statement” [SQL Anywhere Server - SQL Reference]

See also

- “What's new in version 16.0” on page 1

SQL Anywhere Server upgrades

Before using existing applications with this version of the software, be sure to review the list of behavior changes to determine whether your application is affected. See “SQL Anywhere 16 - Changes and Upgrading” on page 1.

Upgrading version 10 and later databases

If you are upgrading from version 10 or later, you can either upgrade and/or rebuild your database. Upgrading or rebuilding is an optional step because the version 16 software can be used with a version 10
or later database. However, to take advantage of all the new features in version 16, you must rebuild your
database. See “The upgrade process for version 10 and later databases” on page 286 and “The rebuild
process for version 10 and later databases” on page 276.

Note
It is recommended that you refresh the materialized views in your database after upgrading your database
server, or after rebuilding or upgrading your database to work with an upgraded database server. See
“Manually refreshing a materialized view” [SQL Anywhere Server - SQL Usage].

Upgrading version 9 and earlier databases
If you are upgrading to version 16 from version 9 or earlier, you must rebuild the database, which consists
of unloading the old database, and reloading it into a new version 16 database. Attempting to start version
9 or earlier databases results in an error on database startup. There are several approaches for rebuilding
existing databases:

● Use the version 16 Unload utility (dbunload) with the -an (create a new database) or -ar (replace the old
database) option. See “Rebuild a version 9 or earlier database using the Unload utility (command
line)” on page 285.

Note
The Unload utility (dbunload) has the same file name in all versions of SQL Anywhere. Ensure that
you are using the correct version. Run the command dbunload -? to determine which version of the
Unload utility you are using. See “How to ensure that you are running the correct version of the utilities
when you have multiple versions installed” on page 275.

● Unload the database using the version 16 Unload utility, and then reload the database using the
reload.sql file on the version 16 database server.
If you need to make schema changes, this is the recommended way of upgrading. After you make the
schema changes, you can create a database, and then apply the reload script to it.

● Use the Unload Database Wizard in Sybase Central. You can choose to create a new database, replace
an existing database with the new database, or unload the database to a file. See “Rebuild a version 9 or
earlier database (Sybase Central)” on page 284.

● Unload the database using an older version of dbunload, and then reload the database using the
reload.sql file and the version 16 database server. Only use this approach if the other methods fail
because deprecated or unsupported database option settings, objects, or SQL syntax could be unloaded
into the reload.sql file. If problems occur during the reload, edit the file manually. The internal reload
capabilities of version 16 take care of many of these problems.
Note
SQL Anywhere 9.0.2 for Mac OS X was supported on PPC, while SQL Anywhere 10 and later for Mac OS X are supported on Intel. If you have a version 9.0.2 or earlier database on Mac OS X, you have two options for unloading the database:

● Unload the database using the version 9.0.2 software.

● Copy the database to a different platform where SQL Anywhere 16 is installed, and then unload the database using the version 16 software.

Once the database is unloaded, you can perform the reload on Mac OS X using the version 16 software.

The procedure is more involved to change the characteristics of the database during unload and reload (for example, change a case-sensitive database to a case-insensitive database). See “Database rebuilds” [SQL Anywhere Server - SQL Usage].

Upgrade and rebuild precautions

There are several precautions that you should take before upgrading SQL Anywhere:

● Back up your JRE directory  Upgrading may overwrite the JRE directory (%SQLANY16%\binXX\jre170) and its subdirectories. In this case, if you are using certificates, your certificate store (%SQLANY16%\binXX\jre170\lib\security\cacerts) is overwritten, including your certificates. Similarly, fonts you added to the %SQLANY16%\binXX\jre170\lib\fonts\fallback directory to help display characters in the administration tools may be lost. To minimize upgrading steps with regard to the JRE change, create a backup copy of the JRE directory and all of its subdirectories before you upgrade so that you can refer to or restore files (such as cacerts) from the backup, as needed.

● Check the behavior changes  Confirm that none of the documented behavior changes affect your application. If they do, update your application. See “What's new in version 16.0” on page 1.

● Test your application  Test your application thoroughly in a SQL Anywhere 16 environment before upgrading any applications in production use.

● Use the correct version of the utilities  Make sure that you use the correct version of the database utilities with your new database. See “How to ensure that you are running the correct version of the utilities when you have multiple versions installed” on page 275.

● Validate and back up the database  Before you begin an upgrade, validate your database, and back up your software and database. To ensure future recoverability, back up the database when you finish the upgrade.

● Synchronize before upgrading  For databases involved in synchronization, such as UltraLite databases or SQL Anywhere remote databases in MobiLink installations, perform a successful synchronization before upgrading.

● Test your upgrade procedure  Test your upgrade procedure carefully before carrying it out on a production system.
SQL Anywhere is used in many different configurations, and no upgrade guidelines can be guaranteed for all cases.

**How to ensure that you are running the correct version of the utilities when you have multiple versions installed**

If you have multiple versions of SAP Sybase SQL Anywhere on your Windows computer, pay attention to your system path when using utilities. Since the installation adds the most recently installed version executable directory to the end of your system path, it is possible to install a new version of the software, and still inadvertently be running the previously installed version.

For example, if a version 8 executable directory is ahead of the 16 executable directory in your path and you use the dbinit command, you use the version 8 utility, and consequently create a version 8 database.

There are several ways you can ensure that you are using the version 16 utilities, including:

- Modify your system path so that the version 16 executable directory is before any previous version executable directory.
- Change to the version 16 executable directory before running your command.
- Specify a fully-qualified path name to the utility name that indicates the exact location of the utility you want to run.
- Create scripts to change your environment to use the correct version of the utilities.
- Uninstall the old software.

**Rebuilding (unloading/reloading) a database**

For previous users of the software, this task summarizes the process for rebuilding your database to version 16.

**Prerequisites**

For version 16 databases, you must have the following system privileges:

- BACKUP DATABASE
- VALIDATE ANY OBJECT
- SERVER OPERATOR
- SELECT ANY TABLE
Follow the standard precautions for upgrading software. See “Upgrade and rebuild precautions” on page 274.

Back up the database. For example:

```
   dbbackup -c "DBF=mydb.db;UID=DBA;PWD=sql" old-db-backup-dir
```

**Rebuild a database (command line)**

1. If possible, defragment the drive where the new database will be stored because a fragmented drive can decrease database performance.

2. Shut down all SQL Anywhere database servers because the version 16 dbunload utility cannot be used against a database that is running on a previous version of the database server. For example:

   ```
   dbstop -c "DBF=mydb.db;UID=DBA;PWD=sql"
   ```

3. Unload and reload (rebuild) the old database into a new version 16 database. For example:

   ```
   dbunload -c "DBF=mydb.db;UID=DBA;PWD=sql" -an mydb16.db
   ```

4. Back up the new database before using it. For example:

   ```
   dbbackup -c "DBF=mydb16.db;UID=DBA;PWD=sql" new-db-backup-dir
   ```

5. Validate the new database before using it. For example:

   ```
   dbvalid -c "DBF=mydb16.db;UID=DBA;PWD=sql"
   ```

**Results**

The database is rebuilt to the latest version. By default, the database is stopped and restarted.

**Next**

You can test the rebuilt database with your application.

**See also**

- “Rebuild a version 9 or earlier database using the Unload utility (command line)” on page 285
- “Rebuild a version 9 or earlier database (Sybase Central)” on page 284
- “Upgrading authenticated databases” [SQL Anywhere Server - Database Administration]

**The rebuild process for version 10 and later databases**

Rebuilding a database consists of unloading and reloading the database to upgrade its file format. When you upgrade the file format, it changes the format used to store and access data on disk, letting you use all the new features and performance enhancements in the latest version of the software.
Caution
Unloading and reloading a large database can be time consuming and can require a large amount of disk space. The process may require disk space approximately twice the size of your database to hold the unloaded data and the new database file.

Because of index changes in SQL Anywhere, when you rebuild a database by unloading and reloading it, the rebuilt database may be smaller than the original database. This decrease in database size does not indicate a problem or a loss of data.

Note
It is recommended that you back up your database before you rebuild it.

Reloading tables with AUTOINCREMENT columns
You can retain the next available value for AUTOINCREMENT columns in the rebuilt database by specifying the dbunload -l option. This option adds calls to the sa_reset_identity system procedure to the generated reload.sql script for each table that contains an AUTOINCREMENT value, preserving the current value of SYSTABCOL.max_identity.

Rebuilding a database (Sybase Central)
You can use the Unload Database Wizard to upgrade a version 10 or later SQL Anywhere database to the latest version.

Prerequisites
Follow the standard precautions for upgrading software. See “Upgrade and rebuild precautions” on page 274.

You must have the SELECT ANY TABLE and the SERVER OPERATOR system privileges.

Ensure that the database is backed up before unloading and reloading it. For example:

```
dbbackup -c "DBF=mydb.db;UID=DBA;PWD=sql" old-db-backup-dir
```

Rebuild a database (Sybase Central)
1. Click Start » Programs » SQL Anywhere 16 » Administration Tools » Sybase Central.
2. Start a version 16 database server running the database you want to upgrade, and then connect to the database from Sybase Central.
3. Click Tools » SQL Anywhere 16 » Unload Database.
4. Read the text on the first page of the Unload Database Wizard and then click Next.
5. Click Unload a database running on a current version of the server, and then select the database from the list. Click Next.
6. Choose to unload and reload into a new database. Click **Next**.

7. Specify a new file name for the database.

8. You can also specify the page size for the new database, but the page size you specify cannot be larger than the database server page size. The default page size is 4096 bytes. You can encrypt the database file if you want. If you choose strong encryption, you need the encryption key each time you want to start the database. Click **Next**.

9. Click **Unload structure and data**. You can also select any other options you want for your database. Click **Next**.

10. Click **Unload all database objects**. Click **Next**.

11. Specify whether you want to connect to the new database when the unload/reload is complete.

12. Click **Finish** to start the process.

**Results**

The database is upgraded to the latest version. By default, the database is stopped and restarted.

**Next**

Examine the new database to confirm that the rebuild completed properly and test the upgraded database with your application.

**See also**

- “Backing up databases” [*SQL Anywhere Server - Database Administration*]
- “Tips on exporting data with the Unload Database Wizard” [*SQL Anywhere Server - SQL Usage*]
- “Database rebuilds” [*SQL Anywhere Server - SQL Usage*]
- “Database encryption and decryption” [*SQL Anywhere Server - Database Administration*]

**Rebuilding a database (command line)**

You can use the dbunload utility to upgrade a version 10 or later SQL Anywhere database to the latest version.

**Prerequisites**

The database user specified in the connection-string must have the SELECT ANY TABLE and SERVER OPERATOR system privileges.

Follow the standard precautions for upgrading software. See “Upgrade and rebuild precautions” on page 274.

Ensure that you have exclusive access to the database being upgraded and ensure that the path of the version 16 utilities is ahead of the path of the other utilities in your system path. See “How to ensure that
you are running the correct version of the utilities when you have multiple versions installed” on page 275.

Ensure that the database is backed up before unloading and reloading. For example:

```
dbbackup -c "DBF=mydb.db;UID=DBA;PWD=sql" old-db-backup-dir
```

**Context and remarks**

When using dbunload with a version 10 or later database, the version of dbunload used must match the version of the database server used to access the database. If an older version of dbunload is used with a newer database server, or vice versa, an error is returned.

If you are rebuilding a database that is a remote database in a MobiLink installation or that is involved in SQL Remote replication, and if you use the dbunload utility, you must be sure to use the -ar or -an option. These options ensure that the transaction log offsets for the new database are set to match those of the old database.

**Rebuild a database (command line)**

1. Run the Unload utility (dbunload) and use the -an option to create a database.

```
dbunload -c "connection-string" -an new-db-file
```

This command creates a database. To replace the existing database with an upgraded database, use the -ar option instead of -an. To use the -ar option, connect to a personal database server, or to a network database server on the same computer as the Unload utility (dbunload).

For information about other Unload utility (dbunload) options, see “Unload utility (dbunload)” [SQL Anywhere Server - Database Administration].

2. Shut down the database and archive the transaction log before using the reloaded database.

To change the characteristics of the database during unload and reload (for example, change a case-sensitive database to a case-insensitive database), the procedure is more involved. See “Database rebuilds” [SQL Anywhere Server - SQL Usage].

**Results**

The database is upgraded to the latest version. By default, the database is stopped and restarted.

**Next**

Examine the new database to confirm that the rebuild completed properly and test the upgraded database with your application.

**See also**

- “Backing up databases” [SQL Anywhere Server - Database Administration]
- “Unload utility (dbunload)” [SQL Anywhere Server - Database Administration]
- “Database rebuilds” [SQL Anywhere Server - SQL Usage]
The rebuild process for version 9 and earlier databases

It is recommended that you back up your database before rebuilding it.

Note
The page size for a database can be (in bytes) 2048, 4096, 8192, 16384, or 32768, with the default being the page size of the original database.

For information about upgrading Windows Mobile databases, see “Database rebuilding on Windows Mobile” [SQL Anywhere Server - Database Administration].

Note
SQL Anywhere 9.0.2 for Mac OS X was supported on PPC, while SQL Anywhere 10.0.0 and later for Mac OS X are supported on Intel. If you have a version 9.0.2 or earlier database on Mac OS X, you have two options for unloading the database:

- Unload the database using the version 9.0.2 software.
- Copy the database to a different platform where SQL Anywhere 16 is installed, and then unload the database using the version 16 software.

Once the database is unloaded, you can perform the reload on Mac OS X using the version 16 software.

Caution
Unloading and reloading a large database can be time consuming and can require a large amount of disk space. The process requires access to disk space twice the size of your database to hold the unloaded data and the new database file.

Upgrade restrictions
There are some restrictions to note when rebuilding version 9 or earlier databases using the version 16 tools:

- Disconnect the database from any earlier versions of the database server, and shut down any earlier database servers running on the computer. You must also shut down any version 16 database servers that are running on the computer. If dbunload cannot proceed because it detects any of these cases, it issues an error and fails.

- Do not include the ENG, START, or LINKS connection parameters in the dbunload connection string for the old database (specified in the -c option). If you specify these parameters, they are ignored and a warning appears. In the Sybase Central Connect window, do not enter values in the Server name or Start line fields.

- Run dbunload on a computer with direct file system access to the old database (dbunload must be able to connect to the database using shared memory).
Do not run a database server named dbunload_support_engine on the computer where the rebuild is taking place.

Do not unload a version 9 or earlier database when the database file requires recovery. When you use the Unload utility (dbunload) on a database file that requires recovery, a message is returned indicating that the database could not be started. Use a version 9 database server to start the database and then stop the database before retrying dbunload. To unload a version 9 or later database, you must be able to start the database in read-only mode. See “-r database option” [SQL Anywhere Server - Database Administration].

Special considerations

**Password case sensitivity** In newly-created SQL Anywhere 16 databases, all passwords are case sensitive, regardless of the case-sensitivity of the database. The default DBA password for new databases is sql.

When you rebuild an existing database, SQL Anywhere determines the case sensitivity of the password as follows:

- If the password was originally entered in a case-insensitive database, the password remains case-insensitive.
- If the password was originally entered in a case-sensitive database, uppercase and mixed case passwords remain case sensitive. However, if the password was entered in all lowercase, then the password becomes case-insensitive.
- Changes to both existing passwords and new passwords are case sensitive.

**Page sizes** The default database page size for SQL Anywhere 16 databases is 4096 bytes. The supported page sizes in version 16 are 2048 bytes, 4096 bytes, 8192 bytes, 16384 bytes, and 32768 bytes. If your old database uses an unsupported page size, the new database has a page size of 4096 bytes by default. You can use the dbinit -p option or the dbunload -ap option to specify a different page size. See “Initialization utility (dbinit)” [SQL Anywhere Server - Database Administration] and “Unload utility (dbunload)” [SQL Anywhere Server - Database Administration].

**Collations** In version 9 and earlier, SQL Anywhere supported one collation used with CHAR data types. This collation used the SQL Anywhere Collation Algorithm (SACA). In version 10 and later, SQL Anywhere supports two collation algorithms, SACA and UCA (Unicode Collation Algorithm). Unless you specify a new or different collation for the rebuilt database, the SACA collation from the old database is unloaded and reused in the rebuilt database.

If you are rebuilding a database with a custom collation, the collation is preserved only if you rebuild in a single step (internal unload). If you choose to unload the database, and then load the schema and data into a database that you create, then you must use one of the supplied collations. See “Alternate collations” [SQL Anywhere Server - Database Administration].

**Database file size** Because of index changes in SQL Anywhere, when you rebuild a database by unloading and reloading it, the rebuilt database may be smaller than the original database. This decrease in database size does not indicate a problem or a loss of data.
**Known issues**

If the rebuild process fails when you run dbunload or the **Unload Database Wizard**, you can use the following steps to help diagnose the reason for the failure.

1. Create a new, empty version 16 database.
   
   ```bash
   dbinit -dba DBA,sql test.db
   ```

2. Apply the `reload.sql` file to the empty database.
   
   ```bash
   dbisql -c "DBF=test.db;UID=DBA;pwd=sql" reload.sql
   ```

3. Change the `reload.sql` file or the original database based on the messages you receive when applying the `reload.sql` file to the new database.

The following table lists issues that are known to cause a rebuild to fail, as well as their solutions.

<table>
<thead>
<tr>
<th>Known problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>A DECLARE LOCAL TEMPORARY TABLE statement in a procedure or trigger causes a</td>
<td>Remove the owner name.</td>
</tr>
<tr>
<td>syntax error if the table name is prefixed with an owner name.</td>
<td></td>
</tr>
<tr>
<td>If a CREATE TRIGGER statement does not include an owner name for the table</td>
<td>Prefix the table name with the owner name.</td>
</tr>
<tr>
<td>on which the trigger is defined, and the table must be qualified with an</td>
<td></td>
</tr>
<tr>
<td>owner when referenced by the user executing the <code>reload.sql</code> file, the</td>
<td></td>
</tr>
<tr>
<td>statement fails and an error is returned indicating that the table could</td>
<td></td>
</tr>
<tr>
<td>not be found.</td>
<td></td>
</tr>
<tr>
<td>If an object name (such as a table, column, variable, or parameter name)</td>
<td>Change all references to the reserved word to use a different name. For</td>
</tr>
<tr>
<td>corresponds to a reserved word introduced in a later version of SQL Anywhere</td>
<td>variable names, prefixing the name with @ is a common convention that</td>
</tr>
<tr>
<td>(such as NCHAR), then the reload fails. For example:</td>
<td>prevents naming conflicts.</td>
</tr>
<tr>
<td>CREATE PROCEDURE p( )</td>
<td>For a complete list of reserved words, see <strong>“Reserved words”</strong> [SQL</td>
</tr>
<tr>
<td>BEGIN</td>
<td>Anywhere Server - SQL Reference].</td>
</tr>
<tr>
<td>DECLARE NCHAR INT;</td>
<td></td>
</tr>
<tr>
<td>SET NCHAR = 1;</td>
<td></td>
</tr>
<tr>
<td>END;</td>
<td></td>
</tr>
</tbody>
</table>
### Known problem

<table>
<thead>
<tr>
<th>Known problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>If a database is unloaded with a version 9 or earlier copy of dbunload, the <code>reload.sql</code> file can contain calls to the ml_add_property system procedure, but this procedure is not present in a new version 16 database.</td>
<td>Unload the database with the version 16 dbunload utility. For information about ensuring you are using the correct version of the database utilities, see “How to ensure that you are running the correct version of the utilities when you have multiple versions installed” on page 275.</td>
</tr>
<tr>
<td>If you unload a database using a version 9 or earlier version of dbunload, views that use Transact-SQL outer joins (by specifying <em>= or =</em>) may not be created properly when they are reloaded.</td>
<td>Add the following line to the reload script: <code>SET TEMPORARY OPTION tsql_outer_joins='on';</code> You should later rewrite any views that use Transact-SQL outer joins.</td>
</tr>
<tr>
<td>The [NOT] DETERMINISTIC clause is not supported in the CREATE PROCEDURE and ALTER PROCEDURE statements. If the clause is present, the reload fails and a syntax error is returned.</td>
<td>If you are upgrading a database that contains user-defined procedures that include the [NOT] DETERMINISTIC clause, you must remove the clause before you unload and reload the database.</td>
</tr>
</tbody>
</table>

---

**Rebuilding a version 9 or earlier database (Sybase Central)**

The **Unload Database Wizard** rebuilds a version 9 or earlier database by unloading into a reload file and data files, unloading and reloading into a new database, or unloading and reloading into an existing database. Rebuilt databases support all new features and performance enhancements in the latest software version.

**Prerequisites**

You must have the SELECT ANY TABLE and SERVER OPERATOR system privileges.

It is recommended that you back up your database before rebuilding it.

The database file must be located on the same computer as the SQL Anywhere 16 installation.

You cannot unload a subset of tables from a database. Use the dbunload utility to do this.

If the **Unload Database Wizard** determines that the database file is already running, then the database is stopped before the unload proceeds.

Follow the standard precautions for upgrading software. See “Upgrade and rebuild precautions” on page 274.

If possible, defragment the drive where the new database will be stored because a fragmented drive can decrease database performance.
Ensure that you have exclusive access to the database to be unloaded and reloaded. No other users can be connected.

**Rebuild a version 9 or earlier database (Sybase Central)**

1. Click **Start** » **Programs** » **SQL Anywhere 16** » **Administration Tools** » **Sybase Central**.

2. Click **Tools** » **SQL Anywhere 16** » **Unload Database**.

3. Read the introductory page of the **Unload Database Wizard**, and click **Next**.

4. Click **Unload a database running on an earlier version of the server, or a database that is not running**. Specify the connection information for the database. Click **Next**.

5. Click **Unload and reload into a new database**. Click **Next**.

6. Specify a new file name for the database. Click **Next**.

   You can specify the page size for the new database. In version 16, the default (and recommended) page size is 4096 bytes.

   You can encrypt the database file. If you choose strong encryption, you need the encryption key each time you start the database. See “Database encryption and decryption” [SQL Anywhere Server - Database Administration].

7. Choose to unload structure and data. Click **Next**.

8. Specify whether you want to connect to the new database when the rebuild is complete.

9. Click **Finish**. Examine the new database to confirm that the rebuild completed properly.

**Results**

The database is upgraded to the latest version. By default, the database is stopped and restarted.

**Next**

Examine the new database to confirm that the rebuild completed properly and test the rebuilt database with your application.

**See also**

- “Backing up databases” [SQL Anywhere Server - Database Administration]
- “Unload utility (dbunload)” [SQL Anywhere Server - Database Administration]
- “Database rebuilds” [SQL Anywhere Server - SQL Usage]
Rebuilding a version 9 or earlier database using the Unload utility (command line)

You can use the Unload utility (dbunload) -an or -ar option to rebuild a version 9 or earlier database. Rebuilt databases support all new features and performance enhancements in the latest software version.

Prerequisites

The database user specified in the connection-string must have the SELECT ANY TABLE and SERVER OPERATOR system privileges.

It is recommended that you back up your database before rebuilding it.

The database file must be located on the same computer as the SQL Anywhere 16 installation.

Follow the standard precautions for upgrading software. See “Upgrade and rebuild precautions” on page 274.

Ensure that you have exclusive access to the database to be unloaded and reloaded. No other users can be connected.

Ensure that the version 16 utilities are ahead of other utilities in your system path. See “How to ensure that you are running the correct version of the utilities when you have multiple versions installed” on page 275.

Context and remarks

The -an option is recommended because it creates a new database leaving the original database intact. The -ar option replaces your old database with a new version 16 database.

Rebuild a version 9 or earlier database using the Unload utility (command line)

1. Shut down all SQL Anywhere and Adaptive Server Anywhere database servers because the version 16 dbunload utility cannot be used against a database that is running on a previous version of the database server. For example:

   `dbstop -c "DBF=mydb.db;UID=DBA;PWD=sql"

2. If possible, defragment the drive where the new database will be stored because a fragmented drive can decrease database performance.

3. Back up the database. For example:

   `dbackup -c "DBF=mydb.db;UID=DBA;PWD=sql" old-db-backup-dir`

   You must have the BACKUP DATABASE system privilege. See “Backing up databases” [SQL Anywhere Server - Database Administration].

4. Run the Unload utility (dbunload) using the -an or -ar option to create a new database.

   `dbunload -c "connection-string" -an database-filename`
For example:

dbunload -c "DBF=mydb.db;UID=DBA;PWD=sql" -an mydb16.db

This command creates a database (by specifying -an). If you specify the -ar option, the existing database is replaced with a rebuilt database. To use the -ar option, you must connect to a personal database server or to a network database server on the same computer as the Unload utility (dbunload).

Results

The database is upgraded to the latest version. By default, the database is stopped and restarted.

Next

Examine the new database to confirm that the rebuild completed properly and test the rebuilt database with your application.

See also

● “Unload utility (dbunload)” [SQL Anywhere Server - Database Administration]

The upgrade process for version 10 and later databases

Upgrading a database adds and modifies system tables, system procedures, and database options to enable version 16 features. It does not change the file format used to store and access data on disk, and so does not give access to all new features and performance enhancements in the latest version of the software.

For information about upgrading the database file format, see “The rebuild process for version 10 and later databases” on page 276.

Upgrading a version 10 or later database (Sybase Central)

You can use the Upgrade Database Wizard to upgrade a version 10 or later database by adding and modifying system tables, system procedures, and database options to enable version 16 features.

Prerequisites

You must have the following system privileges:

● ALTER DATABASE

● BACKUP DATABASE

● SERVER OPERATOR

Follow the standard precautions for upgrading software. See “Upgrade and rebuild precautions” on page 274.
Caution
Back up your database files before upgrading. If you apply the upgrade to the existing files, then these files become unusable if the upgrade fails. To back up your database, you must have the BACKUP DATABASE system privilege. For information about backing up your database, see “Backup and data recovery” [SQL Anywhere Server - Database Administration].

Context and remarks

The Upgrade Database Wizard does not upgrade a version 9.0.2 or earlier database to version 16. To upgrade an existing version 9.0.2 or earlier database to version 16, you must unload and reload the database using dbunload or the Unload Database Wizard. See “Upgrading version 9 and earlier databases” on page 273.

Upgrade a database (Sybase Central)

1. Follow the standard precautions for upgrading software. See “Upgrade and rebuild precautions” on page 274.

2. Click Start » Programs » SQL Anywhere 16 » Administration Tools » Sybase Central.

3. From the SQL Anywhere 16 plug-in, connect to the database you want to upgrade. The database must be running on a version 16 database server.

4. Click Tools » SQL Anywhere 16 » Upgrade Database.

5. Follow the instructions in the Upgrade Database Wizard. By default, the database is stopped and restarted.

6. (Optional) Stop the database and archive the transaction log by making a copy of it before using the upgraded database if you did not choose to do so in the wizard.

Tip
You can also access the Upgrade Database Wizard by:

- Right-clicking a database, and clicking Upgrade Database.
- Selecting a database, and clicking File » Upgrade Database.

Results

The database is upgraded and contains new and modified system tables, system procedures, and database options.

Next

Examine the new database to confirm that the upgrade completed properly and test the upgraded database with your application.
See also

- “Upgrade utility (dbupgrad)” [SQL Anywhere Server - Database Administration]

Upgrading a version 10 or later database (command line)

You can use the dbupgrad utility to upgrade a version 10 or later database by adding and modifying system tables, system procedures, and database options to enable version 16 features.

Prerequisites

The database user specified in the connection-string must have the ALTER DATABASE system privilege, and must be the only connection to the database.

Follow the standard precautions for upgrading software. See “Upgrade and rebuild precautions” on page 274.

Ensure that you have exclusive access to the database to be upgraded and ensure that the version 16 utilities are ahead of other utilities in your system path. See “How to ensure that you are running the correct version of the utilities when you have multiple versions installed” on page 275.

Caution

Back up your database files before upgrading. If you apply the upgrade to the existing files, then these files become unusable if the upgrade fails. For information about backing up your database, see “Backup and data recovery” [SQL Anywhere Server - Database Administration].

Upgrade a database (command line)

1. Run the Upgrade utility (dbupgrad) against the database:

   `dbupgrad -c "connection-string"

   For more information, see “Upgrade utility (dbupgrad)” [SQL Anywhere Server - Database Administration].

2. Shut down the database and archive the transaction log before using the upgraded database.

Results

The database is upgraded and contains new and modified system tables, system procedures, and database options.

Next

Examine the new database to confirm that the upgrade completed properly and test the upgraded database with your application.

See also

- “Upgrade utility (dbupgrad)” [SQL Anywhere Server - Database Administration]
Upgrading a version 10 or later database (SQL)
You can use the ALTER DATABASE statement to upgrade a version 10 or later database by adding and modifying system tables, system procedures, and database options to enable version 16 features.

Prerequisites
Follow the standard precautions for upgrading software. See “Upgrade and rebuild precautions” on page 274.

You must be the owner of the database or have ALTER DATABASE system privilege, and must be the only connection to the database.

Caution
Back up your database files before upgrading. If you apply the upgrade to the existing files, then these files become unusable if the upgrade fails.

Upgrade a database (SQL)
1. Connect to the database from Interactive SQL or another application that can execute SQL statements. No other connection can be using the database at the same time.

2. Execute an ALTER DATABASE statement.
   For example, the following statement upgrades a database:
   
   ```sql
   ALTER DATABASE UPGRADE;
   ```

3. Shut down the database and archive the transaction log before using the upgraded database.

Results
The database is upgraded and contains new and modified system tables, system procedures, and database options.

Next
You should examine the new database to confirm that the upgrade completed properly and test the upgraded database with your application.

See also
- “ALTER DATABASE statement” [SQL Anywhere Server - SQL Reference]
- “Upgrade and rebuild precautions” on page 274
- “Backup and data recovery” [SQL Anywhere Server - Database Administration]
Upgrades and rebuilds in a database mirroring system

All database servers in a database mirroring system must use the same minor release.

Updating the database server software to a major version for a mirroring system without rebuilding the database

Update the software of the database servers in a mirroring system by installing the software, stopping each database server, and starting the databases on servers running the new software. The database itself is not upgraded or rebuilt.

Prerequisites

You must have the BACKUP DATABASE system privilege. You must be the owner of the database, or have the VALIDATE ANY OBJECT system privilege.

By default, you must have the SERVER OPERATOR system privilege to stop network database servers.

Context and remarks

Test the following steps in non-production environment with your application before performing the steps in a production environment.

All database servers in a database mirroring system must use the same minor release of SQL Anywhere. So, the mirroring system stops temporarily during the update.

Apply a SQL Anywhere major version to a database mirroring system

1. Make a backup of the primary database, copy the backup, and validate the copy of the backup.

   For example, the following command backs up a database named mydb.db:

   ```
   dbbackup -c "DBN=mydb;ENG=myserver;UID=DBA;PWD=sql" backup-dir
   ```

   To create a copy of the backup and apply the transaction logs to the copy of the backup, run the following commands:

   ```
   xcopy backup-dir\mydb.db validatebackup-dir
   xcopy backup-dir\mydb.log validatebackup-dir
   ```

   Validate this copy.

   ```
   dbvalid -c "DBF=validatebackup-dir\backupmydb.db;UID=DBA;PWD=sql"
   ```

   If the copy of the backup does not validate, then fix the problems that cause the validation to fail before proceeding. Otherwise, you risk losing data.
2. Install the new software on the primary, mirror, and arbiter servers.

3. If the mirroring system is involved in read-only scale-out, install the software on the copy nodes.

4. Stop the servers in the following order:
   a. copy nodes
   b. mirror server
   c. primary server
   d. arbiter server

   For example, run the Stop utility (dbstop):

   `dbstop -y -c "UID=DBA;PWD=sql;Server=myserver"

5. (Optional) Upgrade or rebuild the databases.

6. Start the databases on the new servers in the following order:
   a. arbiter server, primary server, and mirror server
   b. copy nodes

**Results**

The databases in the mirroring system run on the new version of the software.

**Next**

Examine the database mirroring system to confirm that the update completed properly, and test the database mirroring system with your application.

**See also**

- “Stopping a database server in a mirroring system” [SQL Anywhere Server - Database Administration]
- “User initiated role switches (failovers)” [SQL Anywhere Server - Database Administration]
- “Stop Server utility (dbstop)” [SQL Anywhere Server - Database Administration]

**Updating the database server software to a different minor release for a mirroring system without rebuilding the database**

Update the software of the database servers in a mirroring system by installing the software, stopping each server, and starting the database on the new server. All database servers in a database mirroring system must use the same minor release.
Prerequisites

You must have the BACKUP DATABASE system privilege. You must be the owner of the database, or have the VALIDATE ANY OBJECT system privilege.

By default, you must have the SERVER OPERATOR system privilege to stop network database servers.

Test the following steps in non-production environment with your application before performing the steps in a production environment.

Context and remarks

The mirroring system stops temporarily during the update.

Apply a SQL Anywhere major or maintenance version to a database mirroring system

1. Make a backup of the primary database, copy the backup, and validate the copy of the backup.

   For example, run the following command to back up a database named mydb.db:
   
   `dbbackup -c "DBN=mydb;ENG=myserver;UID=DBA;PWD=sql" backup-dir`

   Create a copy of the backup:

   `xcopy backup-dir\*.db validatebackup-dir\backupmydb.db`

   Validate the backup copy:

   `dbvalid -c "DBF=validatebackup-dir\backupmydb.db;UID=DBA;PWD=sql"`

   If the copy of the backup does not validate, then fix the problems that cause the validation to fail before proceeding. Otherwise, you risk losing data.

2. Stop the servers in the following order:

   a. copy nodes
   b. mirror server
   c. primary server
   d. arbiter server

   For example, run the Stop utility (dbstop):

   `dbstop -y -c "UID=DBA;PWD=sql;Server=myserver"`

3. Install the new software on the primary, mirror, and arbiter servers.

4. Start the servers in the following order:

   a. arbiter server
   b. primary server
   c. mirror server
5. If the mirroring system is involved in read-only scale-out, install the software on the copy nodes, and then restart the copy nodes

Results

The databases in the mirroring system run on the new version of the software.

Next

Examine the database mirroring system to confirm that the upgrade completed properly, and test the database mirroring system with your application.

See also

- “Stopping a database server in a mirroring system” [SQL Anywhere Server - Database Administration]
- “User initiated role switches (failovers)” [SQL Anywhere Server - Database Administration]
- “Stop Server utility (dbstop)” [SQL Anywhere Server - Database Administration]

Updating the database server software to a Support Package for database mirroring system without rebuilding the database

Update the software of the database servers in a mirroring system by stopping each server, installing the software, and then restarting the database on the new server.

Prerequisites

You must have the BACKUP DATABASE system privilege. You must be the owner of the database, or have the VALIDATE ANY OBJECT system privilege.

By default, you must have the SERVER OPERATOR system privilege to stop network database servers.

Context and remarks

Because you can stop servers in the system one at a time, the mirroring system can continue to run during a Support Package update of the server software. A failover from the primary server to the mirror server occurs at least once during the update process. Connections to the primary and mirror server drop during any failover.

Test the following steps in non-production environment with your applications before performing the steps in a production environment.

Apply a Support Package to a database mirroring system

1. Make a backup of the primary database, copy the backup, and validate the copy of the backup.

   For example, run the following command to back up a database named mydb.db:
Create a copy of the backup:

```
xcopy backup-dir\*.db validatebackup-dir\backupmydb.db
```

Validate the backup copy:

```
dbvalid -c "DBF=validatebackup-dir\backupmydb.db;UID=DBA;PWD=sql"
```

If the copy of the backup does not validate, then fix the problems that cause the validation to fail before proceeding. Otherwise, you risk losing data.

2. If the mirroring system is part of a read-only scale-out system, then for each copy node:
   a. Stop the copy node server.
   b. Install the software.
   c. Start the copy node.

3. Stop the mirror server.

4. Install the software on the mirror server.

5. Start the mirror server and ensure that it is in a synchronized state.

6. Stop the arbiter server.

7. Install the software on the arbiter.

8. Start the arbiter server.

9. Initiate a fail over by connecting to the primary database and executing the following statement:

   ```
   ALTER DATABASE SET PARTNER FAILOVER;
   ```

   Connections to the primary and mirror drop during the failover. The current primary becomes the mirror.

10. Stop the mirror server.

11. Install the software on the mirror server.

12. Start the mirror server and ensure that it is synchronized.

**Results**

The databases in the mirroring system run on the new version of the software.

**Next**

Examine the database mirroring system to confirm that the upgrade completed properly, and test the database mirroring system with your application.
See also
- “Stopping a database server in a mirroring system” [SQL Anywhere Server - Database Administration]
- “User initiated role switches (failovers)” [SQL Anywhere Server - Database Administration]
- “Stop Server utility (dbstop)” [SQL Anywhere Server - Database Administration]

Upgrading or rebuilding (unloading/reloading) the databases in a database mirroring system

Upgrading or rebuilding the databases in a database mirroring system
Upgrade the primary database, and then copy the upgraded database and transaction log to the mirror. The mirroring system is temporarily stopped. Upgrading databases is generally performed when updating to a major version or minor release of the software.

Prerequisites
You must have the BACKUP DATABASE system privilege. You must be the owner of the database, or have the VALIDATE ANY OBJECT system privilege.

By default, you must have the SERVER OPERATOR system privilege to start or stop network database servers.

To upgrade a database, you must have the ALTER DATABASE system privilege, and must be the only connection to the database.

To rebuild (unload/reload) a database using the Unload utility (dbunload), you must have the SELECT ANY TABLE system privilege. For an unload with a reload, you must also have the SERVER OPERATOR system privilege.

Test the following steps in non-production environment with your application before performing the steps in a production environment.

Apply a SQL Anywhere minor release to a database mirroring system

1. Install the new version of the software on each computer in the system. All database servers in a database mirroring system must use the same maintenance version of SQL Anywhere.

2. Make a backup of the primary database, copy the backup, and validate the copy of the backup.

For example, run the following command to back up a database named mydb.db:

dbackup -c "DBN=mydb;ENG=myserver;UID=DBA;PWD=sql" backup-dir

Create a copy of the backup:

xcopy backup-dir\*.db validatebackup-dir\backupmydb.db

Validate the backup copy:

dbvalid -c "DBF=validatebackup-dir\backupmydb.db;UID=DBA;PWD=sql"
If the copy of the backup does not validate, then fix the problems that cause the validation to fail before proceeding. Otherwise, you risk losing data.

3. Stop the servers in the following order:
   a. copy nodes
   b. mirror
   c. primary

4. | Option                        | Action                                                                 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrade the database</td>
<td>Run the Upgrade utility (dbupgrad) on the primary database. For example:</td>
</tr>
<tr>
<td></td>
<td>dbupgrad -c &quot;UID=DBA;PWD=sql;DBF=C:\primary-database.db&quot;</td>
</tr>
<tr>
<td></td>
<td>The database is upgraded, a new transaction log is created, and the</td>
</tr>
<tr>
<td></td>
<td>database is stopped. You can delete the old transaction logs.</td>
</tr>
<tr>
<td>Rebuild (unload/reload) the</td>
<td>dbunload -c &quot;UID=DBA;PWD=sql;DBF=C:\primary-database.db&quot;</td>
</tr>
<tr>
<td>database</td>
<td></td>
</tr>
</tbody>
</table>

5. Copy the upgraded or rebuilt database and its new transaction log to the mirror server and any scale-out copy nodes.

6. Start the servers in the following order:
   a. primary
   b. mirror
   c. copy nodes

Results

The databases in the mirroring system are upgraded or rebuilt and the mirroring system is running.

Next

Examine the database mirroring system to confirm that the upgrade completed properly, and test the database mirroring system with your application.

See also

- “Stopping a database server in a mirroring system” [SQL Anywhere Server - Database Administration]
- “User initiated role switches (failovers)” [SQL Anywhere Server - Database Administration]
- “Stop Server utility (dbstop)” [SQL Anywhere Server - Database Administration]
Troubleshooting: Database upgrades

This section describes some common problems that may occur when upgrading your database.

Troubleshooting: Ensure that JDBC applications are not running when applying a Support Package

After applying a Support Package, you may find that your JDBC applications stop working, and you receive a message similar to the following:

The sajdbc4.jar build does not match the shared object build.

This message can be returned because either Interactive SQL, Sybase Central, the fast launchers, or your own JDBC applications were running when the Support Package was applied. In this case the Java VM locks any DLLs or shared objects that are loaded by it, but does not lock the JAR files. As a result, applying the Support Package often updates the sajdbc4 and jodbc4 JAR files, but not the accompanying dbjdbc16 and/or dbjodbc16 DLL or shared object. When the JDBC application is restarted, the JDBC JAR file does not match the build of the accompanying DLL or shared object, and a message like the one above is returned.

First try shutting down all JDBC-based applications and then reapplying the Support Package. If reapplying the Support Package does not work, try resolving the problem through the following methods:

- **Check that the Support Package installer properly updated DLLs and shared objects**
  - On Windows, check that the dbjdbc16.dll and dbjodbc16.dll files were properly updated when applying the Support Package.
  - On Linux/Unix, check that the libdbjdbc16.so.1 and libdbjodbc16.so.1 shared objects were properly updated when applying the Support Package.
  - On Mac OS X, check that the libdbjdbc16.dylib and libdbjodbc16.dylib shared objects were properly updated when applying the Support Package.

  **Note**
  There may be both 32-bit and 64-bit versions of the DLLs or shared objects installed on your system. When checking the DLLs or shared objects, you must check those that match the bitness of the JAVA VM, not the bitness of the SQL Anywhere server.

- **Check that there are not multiple copies of the DLLs and shared objects on your system**
  If the DLLs or shared objects have been properly updated, make sure that you do not have multiple copies of the DLLs or shared objects with the same bitness. You have multiple copies of DLLS if you have copied the DLLs or shared objects to the extensions folders of the Java VM to bypass the Java restriction that does not allow DLLs and shared objects to be loaded within multiple class loaders.

- **Check that the JAR files were properly updated**
  If your DLLs or shared objects were properly updated, and there are not multiple copies on your system, then make sure that the various JAR files
were properly updated. To check each JAR file, run the following commands, and ensure that the SQL Anywhere version and build number reported by the JAR file matches the SQL Anywhere version and build number of the Support Package you installed.

To check sajdbc4.jar, run the following command (replacing path with the path to the JAR file):
```
java -cp path\sajdbc4.jar sybase.jdbc4.sqlanywhere.IBuildNum
```

To check jodbc4.jar, run the following command (replacing path with the path to the JAR file):
```
java -cp path\jodbc4.jar ianywhere.ml.jdbcodbc.jdbc4.IBuildNum
```

Once you determine which JAR file, DLL, or shared object does not match the Support Package build number, make sure that the file is not locked by an application, and then reapply the Support Package.

**Troubleshooting: Aggregate functions and outer references**

SQL Anywhere follows SQL/2008 standards for clarifying the use of aggregate functions when they appear in a subquery. These changes affect the behavior of statements written for previous versions of the software: previously valid queries may now produce error messages and result sets may change.

When an aggregate function appears in a subquery, and the column referenced by the aggregate function is an outer reference, the entire aggregate function itself is treated as an outer reference. The aggregate function is computed in the outer query block, not in the subquery, and becomes a constant within the subquery.

The following restrictions apply to the use of outer reference aggregate functions in subqueries:

- The outer reference aggregate function can only appear in subqueries that are in the SELECT list or HAVING clause, and these clauses must be in the immediate outer block.
- Outer reference aggregate functions can only contain one outer column reference.
- Local column references and outer column references cannot be mixed in the same aggregate function.

Some problems related to the new standards can be circumvented by rewriting the aggregate function so that it only includes local references. For example, the subquery (SELECT MAX(S.y + R.y) FROM S) contains both a local column reference (S.y) and an outer column reference (R.y), which is now illegal. It can be rewritten as (SELECT MAX(S.y) + R.y FROM S). In the rewrite, the aggregate function has only a local column reference. The same sort of rewrite can be used when an outer reference aggregate function appears in clauses other than SELECT or HAVING.

**Example 1**

The following query produced valid results in SQL Anywhere 7 and earlier versions:
```
SELECT Name,
       ( SELECT SUM( p.Quantity )
         FROM SalesOrderItems )
FROM Products p WHERE p.ID = 300;
```
In SQL Anywhere 8 and later versions, the same query produces an error message stating that the function or column reference must appear in a GROUP BY clause. The statement is no longer valid because SUM(p.Quantity) is treated as an outer reference and computed in the outer query block. The above query is equivalent to the following query:

```sql
SELECT Name,
SUM( p.Quantity ) AS Z,
(SELECT Z
FROM SalesOrderItems )
FROM Products p WHERE p.ID = 300;
```

However, since the aggregate function is now computed in the outer query block, the outer query block is treated as a grouped query and the column name must appear in a GROUP BY clause in the SELECT list. Therefore, this query is also invalid and produces the same error message. To return the same result set in SQL Anywhere 8 and later versions as the first query did in SQL Anywhere 7 and previous versions, you must use the following query:

```sql
SELECT Name,
p.Quantity * (SELECT COUNT( * ) FROM SalesOrderItems )
FROM Products p WHERE p.ID = 300;
```

**Example 2**

In SQL Anywhere 7 and earlier versions, the following query produced the result 30,716, because SUM(p.Quantity) was computed inside the nested SELECT query block and only p.Quantity was treated as an outer reference.

```sql
SELECT (SELECT FIRST SUM( p.Quantity ) FROM SalesOrderItems ) AS ss
FROM Products p WHERE p.ID = 300;
```

In SQL Anywhere 8 and later versions, the same query returns the result 28, because SUM(p.Quantity) is treated as an outer reference and is computed in the outer query block. In other words, the above query is equivalent to the following query:

```sql
SELECT DT.ss
FROM (SELECT SUM( p.Quantity ) AS asum,
(SELECT FIRST asum FROM SalesOrderItems ) AS ss
FROM Products p WHERE p.ID = 300 ) AS DT;
```

To return the same results in SQL Anywhere 8 and later versions as the first query did in SQL Anywhere 7 and previous versions, you must use the following query:

```sql
SELECT p.Quantity * (SELECT COUNT( * ) FROM SalesOrderItems ) AS ss
FROM Products p WHERE p.ID = 300;
```

**See also**

- “The HAVING clause: Selecting groups of data” [SQL Anywhere Server - SQL Usage]
**MobiLink upgrades**

**Compatibility with existing software**

- Version 16 MobiLink clients are incompatible with versions of the MobiLink server before 16.0.
- The version 16 MobiLink server can be used with clients that are version 11 or later. If you must support earlier clients, keep an earlier version of the MobiLink server to support them.
- Confirm that none of the documented behavior changes affect your application. If they do, update your application. See “SQL Anywhere 16 - Changes and Upgrading” on page 1.

**Upgrade order**

If you are upgrading an existing MobiLink installation, upgrade the components in the following order:

1. Shut down the MobiLink servers.
2. Upgrade the consolidated database.
   
   See “Consolidated database upgrades” on page 300.
3. Upgrade the MobiLink servers.
   
   See “MobiLink server upgrades” on page 308.
4. Start the MobiLink servers.
5. Upgrade the MobiLink clients.
   
   The version 16 MobiLink server can only be used with clients that are version 11 or later. MobiLink clients before version 11 must be upgraded to work with version 16 MobiLink server.
   
   For information about SQL Anywhere remote databases, see “SQL Anywhere MobiLink client upgrades” on page 308.
6. For information about UltraLite applications, see “UltraLite upgrades” on page 311.

Before upgrading, check for behavior changes that may affect you and take standard upgrade precautions.

**See also**

- “MobiLink behavior changes” on page 50
- “Upgrade and rebuild precautions” on page 274

**Consolidated database upgrades**

Before you can use the new MobiLink server with an existing consolidated database, you must run upgrade scripts that install new system objects. The upgrade scripts must be run by the owner of the currently installed MobiLink system tables. You can also use the following methods to update the MobiLink system setup:
In the MobiLink plug-in for Sybase Central, click **MobiLink 16 » Project » Consolidated Databases** and right-click the database name and click **Check MobiLink System Setup**. If your database requires setup or upgrading, you are prompted to continue.

When you use the **Deploy Synchronization Model Wizard**, system setup is checked when you connect to your consolidated database. If your database requires setup or upgrading, you are prompted to continue. See “Synchronization models” [MobiLink - Getting Started].

The MobiLink upgrade scripts for 6.0.x have been removed. If you require this upgrade, contact Technical Support (http://www.sybase.com/support).

### Notes

- Use the `ml_add_missing_dnld_scripts` stored procedure to fix missing `download_cursor` and/or `download_delete_cursor` scripts. Invoking this procedure with a script version name defines the missing `download_cursor` and/or `download_delete_cursor` scripts as ignored scripts for every synchronization table used by the given script version.

- If you have `authenticate_user_hashed` scripts that were created earlier than version 10.0.0, change them to accept VARBINARY(32) instead of VARBINARY(20), using the binary equivalent type of your RDBMS.

### Upgrading a consolidated database (SQL Anywhere 10.0.0 and later)

Before you can use the new MobiLink server with an existing SQL Anywhere consolidated database that is version 10.0.0 or later, run upgrade scripts to install new system objects.

### Prerequisites

You must be the owner of the currently installed MobiLink system tables to run the setup scripts.

The MONITOR system privilege is required to invoke the locking/blocking detection logic.

### Context and remarks

You can also use the following methods to update the MobiLink system setup:

- In the MobiLink plug-in for Sybase Central, click **MobiLink 16 » Project » Consolidated Databases** and right-click the database name and click **Check MobiLink System Setup**. If your database requires setup or upgrading, you are prompted to continue.

- When you use the **Deploy Synchronization Model Wizard**, system setup is checked when you connect to your consolidated database. If your database requires setup or upgrading, you are prompted to continue. See “Synchronization models” [MobiLink - Getting Started].

### Task

1. Upgrade the SQL Anywhere software.
See “Upgrading version 10 and later databases” on page 272.

2. Upgrade the MobiLink system setup by running the appropriate upgrade script for the version you are upgrading from.

The upgrade script is called `upgrade_sa.sql`. It is located under your SQL Anywhere installation in `MobiLink\upgrade\version`, where `version` is the SQL Anywhere version you are upgrading from.

For example, connect to the database in Interactive SQL and execute the following statement:

```
READ "C:\Program Files\SQL Anywhere 16\MobiLink\upgrade\10.0.x\upgrade_sa.sql"
```

**Results**

The consolidated database can now be used with the new MobiLink server.

### Upgrading a consolidated database (Adaptive Server Enterprise, SAP Sybase IQ, Oracle, MySQL, or Microsoft SQL Server)

Before you can use the new MobiLink server with an existing consolidated database, run upgrade scripts to install new system objects.

**Prerequisites**

You must be the owner of the currently installed MobiLink system tables to run the setup scripts.

For ASE, the MobiLink server login ID must have a SELECT privilege on `MASTER..SYSTRANSACTIONS`.

For SAP Sybase IQ, the EXECUTE permission on `SP_IQTRANSACTION` is required by MobiLink server to use snapshot isolation for download.

For Oracle, the RDBMS user that the MobiLink server uses to connect to the consolidated database must be able to use the MobiLink system tables, procedures, and so on, without any qualifiers (for example, `SELECT * from ml_user`). The RDBMS user must also have SELECT privilege on `GV$TRANSACTION, GV$SESSION, GV$LOCK, and DBA_OBJECTS, and EXECUTE privileges on DBMS_UTILITY`. You cannot grant permission directly for the `GV$TRANSACTION, GV$SESSION` and `GV$LOCK` synonyms; you must instead grant permission on the underlying `GV_$TRANSACTION, GV_$SESSION, and GV_$LOCK` dynamic performance views. You must connect as SYS to grant this access.

For Microsoft SQL Server, the RDBMS user that the MobiLink server uses to connect to the consolidated database must have permission to VIEW SERVER STATE, permission to SELECT from `SYS.DATABASES, and permission to SELECT from SYS.DM_TRAN_LOCKS and SYS.PARTITIONS, SYS.SYSPROCESSES.`
Context and remarks

Only upgrade the MobiLink system objects in your Adaptive Server Enterprise, Oracle, MySQL, or Microsoft SQL Server consolidated database if your precious version of the MobiLink server is earlier than version 16.0.

You can also use the following methods to update the MobiLink system setup:

- In the MobiLink plug-in for Sybase Central, click MobiLink 16 » Project » Consolidated Databases and right-click the database name and click Check MobiLink System Setup. If your database requires setup or upgrading, you are prompted to continue.

- When you use the Deploy Synchronization Model Wizard, system setup is checked when you connect to your consolidated database. If your database requires setup or upgrading, you are prompted to continue. See “Synchronization models” [MobiLink - Getting Started].

Task

1. For Adaptive Server Enterprise databases, set the SELECT INTO database option. Execute the following statement in Sybase Interactive SQL:

   ```sql
   USE MASTER
   go
   sp_dboption your-database-name, "SELECT INTO", true
   go
   USE your-database-name
   go
   checkpoint
   go
   ```

2. Run the appropriate upgrade script for the version you are upgrading from.

   The upgrade scripts are called upgrade_XXX.sql, where XXX indicates the RDBMS of your consolidated database. They are located under your SQL Anywhere installation in MobiLink\upgrade\version, where version is the MobiLink version you are upgrading from.

   For example, to upgrade a Microsoft SQL Server database on which the MobiLink system tables from version 9.0.2 have been applied, run the following command:

   ```shell
   osql -S server_name -U user_name -P password -i "C:\Program Files\SQL Anywhere 16\MobiLink\upgrade\9.0.2\upgrade_mss.sql"
   ```

Results

The consolidated database can now be used with the new MobiLink server.

Upgrading an IBM DB2 LUW consolidated database

Before you can use the new MobiLink server with an existing IBM DB2 LUW consolidated database, run upgrade scripts to install new system objects.
**Prerequisites**

You must be the owner of the currently installed MobiLink system tables to run the setup scripts.

For DB2 LUW 10.1, permission to SELECT from SYSIBMADM.MON_LOCKWAIT, SNAPSHOT_APPL_INFO is required by the MobiLink server to invoke the locking/blocking detection logic.

For DB2 LUW 9, permission to SELECT from SYSIBMADM.LOCKWAIT, SNAPSHOT_APPL_INFO is required by the MobiLink server to invoke the locking/blocking detection logic.

**Context and remarks**

Upgrade your IBM DB2 LUW consolidated database if your previous version of the MobiLink server is earlier than version 16.0.

For information about how to run the IBM DB2 LUW setup script, see “IBM DB2 LUW consolidated database” [MobiLink - Server Administration].

You can also use the following methods to update the MobiLink system setup:

- In the MobiLink plug-in for Sybase Central, click MobiLink 16 » Project » Consolidated Databases and right-click the database name and click Check MobiLink System Setup. If your database requires setup or upgrading, you are prompted to continue.

- When you use the Deploy Synchronization Model Wizard, system setup is checked when you connect to your consolidated database. If your database requires setup or upgrading, you are prompted to continue. See “Synchronization models” [MobiLink - Getting Started].

**Task**

1. Locate the IBM DB2 LUW upgrade script.

   The upgrade script is called upgrade_db2.sql and is held in the MobiLink/upgrade/version subdirectory of your SQL Anywhere installation. The version directory refers to the version of MobiLink from which you are upgrading.

2. Copy upgrade_db2.sql and modify the copy. Change the CONNECT statement at the start of the script so it works with the instance you want to connect to. Apply the copied SQL script to the consolidated database.

**Results**

The consolidated database can now be used with the new MobiLink server.
Upgrading a consolidated database (SQL Anywhere earlier than 10.0.0)

Before you can use the new MobiLink server with an existing SQL Anywhere consolidated database that is earlier than version 10.0.0, you must run upgrade scripts to install new system objects.

**Prerequisites**

If you have set up a SQL Anywhere consolidated database but never synchronized with it, then you must run the setup script (not the upgrade script). This step only applies to SQL Anywhere consolidated databases. See “SQL Anywhere consolidated database” [MobiLink - Server Administration].

**Context and remarks**

- Before SQL Anywhere version 10.0.0, MobiLink system tables were owned by dbo. To run the setup scripts for a SQL Anywhere database, you must be logged in to the consolidated database as the owner of the MobiLink system tables. It is not enough to run these scripts as a user with permission to change the tables. To run the upgrade scripts, you can use the SETUSER SQL statement to impersonate dbo. For example:

  SETUSER "dbo";

To upgrade a consolidated database in Sybase Central, you should use the GRANT CONNECT statement to create a password for dbo and then connect as dbo. For example:

  GRANT CONNECT TO dbo IDENTIFIED BY password;

In the latter case, after you have upgraded you should use ALTER USER to remove the dbo password. For example:

  ALTER USER TO dbo IDENTIFIED BY "";

**Task**

1. If you are upgrading a SQL Anywhere consolidated database that is earlier than version 10.0.0, you must first upgrade the database to version 16:
   a. Shut down the database server.
   b. Upgrade the database to version 16.
      For instructions, see:
      - “The rebuild process for version 9 and earlier databases” on page 280
   c. Start the database server, logging in as DBA.
      Log in as DBA to upgrade.

2. Run the appropriate upgrade script for the version you are upgrading from.

   The upgrade script is called upgrade_asa.sql. It is located under your SQL Anywhere installation in MobiLink\upgrade\version, where version is the SQL Anywhere version you are upgrading from.

   To run the upgrade scripts, impersonate the dbo user by using the SETUSER SQL statement.
For example, to upgrade a SQL Anywhere version 9.0.2 consolidated database, connect to the database in Interactive SQL and execute the following statement:

```
SETUSER "dbo";
READ 'C:\Program Files\SQL Anywhere 16\MobiLink\upgrade\9.0.2\upgrade_asa.sql'
```

3. Remove the dbo password. For example:

```
GRANT CONNECT TO "dbo";
```

4. If you are running the MobiLink server as a user other than DBA, grant execute permission for that user on the new MobiLink system objects. Which system objects are new depends on which version you are upgrading from. The following code grants the necessary permissions to all MobiLink system objects. Before executing the code, change the user name my_user to the name of the user who is running the MobiLink server.

```
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_column to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_connection_script to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_database to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_device to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_device_address to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_listening to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_passthrough to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_passthrough_repair to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_passthrough_script to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_passthrough_status to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_primary_server to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_property to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_delivery to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_delivery_archive to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_global_props to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_notifications to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_repository to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_repository_archive to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_repository_staging to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_status_history to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_status_history_archive to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_status_staging to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_agent to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_agent_property to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_agent_staging to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_deployed_task to my_user;
```
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_event to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_event_staging to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_managed_remote to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_notify to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_remote_db_class to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_task to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_task_command to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_task_command_property to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_task_property to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_script to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_script_version to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_scripts_modified to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_sis_sync_state to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_subscription to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_table to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_table_script to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_user to my_user;
GRANT EXECUTE ON dbo.ml_add_column to my_user;
GRANT EXECUTE ON dbo.ml_add_connection_script to my_user;
GRANT EXECUTE ON dbo.ml_add_dnet_connection_script to my_user;
GRANT EXECUTE ON dbo.ml_add_java_connection_script to my_user;
GRANT EXECUTE ON dbo.ml_add_java_table_script to my_user;
GRANT EXECUTE ON dbo.ml_add_lang_conn_script_chk to my_user;
GRANT EXECUTE ON dbo.ml_add_lang_connection_script to my_user;
GRANT EXECUTE ON dbo.ml_add_lang_table_script to my_user;
GRANT EXECUTE ON dbo.ml_add_lang_table_script_chk to my_user;
GRANT EXECUTE ON dbo.ml_add_missing_dnlid_scripts;
GRANT EXECUTE ON dbo.ml_add_passthrough to my_user;
GRANT EXECUTE ON dbo.ml_add_passthrough_repair to my_user;
GRANT EXECUTE ON dbo.ml_add_table_script to my_user;
GRANT EXECUTE ON dbo.ml_add_user to my_user;
GRANT EXECUTE ON dbo.ml_delete_device to my_user;
GRANT EXECUTE ON dbo.ml_delete_device_address to my_user;
GRANT EXECUTE ON dbo.ml_delete_listening to my_user;
GRANT EXECUTE ON dbo.ml_delete_passthrough to my_user;
GRANT EXECUTE ON dbo.ml_delete_passthrough_repair to my_user;
GRANT EXECUTE ON dbo.ml_delete_remote_id to my_user;
GRANT EXECUTE ON dbo.ml_delete_sync_state to my_user;
GRANT EXECUTE ON dbo.ml_delete_sync_state_before to my_user;
GRANT EXECUTE ON dbo.ml_delete_user to my_user;
GRANT EXECUTE ON dbo.ml_lock_rid to my_user;
GRANT EXECUTE ON dbo.ml_qa_add_delivery to my_user;
GRANT EXECUTE ON dbo.ml_qa_add_message to my_user;
GRANT EXECUTE ON dbo.ml_qa_add_status from_client to my_user;
GRANT EXECUTE ON dbo.ml_qa_staged_status for_client to my_user;
GRANT EXECUTE ON dbo.ml_qa_upsert_global_prop to my_user;
GRANT EXECUTE ON dbo.ml_ra_add_agent_id to my_user;
GRANT EXECUTE ON dbo.ml_ra_assign_task to my_user;
GRANT EXECUTE ON dbo.ml_ra_cancel_notification to my_user;
GRANT EXECUTE ON dbo.ml_ra_cancel_task_instance to my_user;
GRANT EXECUTE ON dbo.ml_ra_clone_agent_properties to my_user;
GRANT EXECUTE ON dbo.ml_ra_delete_agent_id to my_user;
Results

The consolidated database can now be used with the new MobiLink server.

MobiLink server upgrades

Upgrade the MobiLink server to version 16 if you are synchronizing version 16 remote databases.

Before using a version 16 MobiLink server, check the behavior changes to see if any affect you. See “SQL Anywhere 16 - Changes and Upgrading” on page 1.

Version 16 of the MobiLink server only supports version 10, 11 and 12 SQL Anywhere and UltraLite clients. If you must support earlier clients, keep an earlier version of the MobiLink server for supporting them.

SQL Anywhere MobiLink client upgrades

In a production environment, only upgrade SQL Anywhere remote databases after you have upgraded both the consolidated database and the MobiLink server.

In version 10.0.0, Adaptive Server Anywhere was renamed to SQL Anywhere.
There are several kinds of upgrade to consider:

- Upgrading the software.
- Upgrading the remote database itself.
- Upgrading the whole application.

### Upgrading the software

It is recommended that you upgrade dbmlsync and the SQL Anywhere database server at the same time. Version 16 MobiLink clients can only synchronize version 16 databases running on version 16 database servers.

Version 16 MobiLink clients require a version 16 or later MobiLink server for synchronization. Version 16 MobiLink clients do not synchronize with a MobiLink server earlier than version 16.

For information about upgrading MobiLink, see “MobiLink upgrades” on page 300.

### Upgrading SQL Anywhere remotes

You can upgrade MobiLink SQL Anywhere remote databases using the procedures for SQL Anywhere databases. For instructions, see “SQL Anywhere Server upgrades” on page 272.

When there is a schema change or other significant database change, you may need to perform a manual unload and reload. See “Unloading/reloading a remote SQL Anywhere database manually” on page 309.

### Upgrading applications

When deploying a new version of a MobiLink application, it is recommended that you use a new script version for the synchronization scripts. For example, if the existing application uses a script version called v1, then the upgraded application could use a script version called v2. Both script versions can be in use at the same time, which makes it easier to upgrade the remote databases incrementally, rather than all at once.

For version 9.0.0 and later, the MobiLink server -zd option has been removed. If your deployment uses the -zd option and you want to upgrade, change your download scripts to accept the last download timestamp as the first parameter. Alternatively, you can upgrade your client and start using named parameters, which enable you to put script parameters in any order.

### Unloading/reloading a remote SQL Anywhere database manually

When there is a schema change or other significant database change, you may need to perform a manual unload and reload.

#### Prerequisites

For version 16 databases, you must have the following system privileges:

- BACKUP DATABASE
Follow the standard precautions for upgrading software. See “Upgrade and rebuild precautions” on page 274.

**Task**

1. Stop all database activity.

2. Perform a successful synchronization and validate and back up the remote database.

3. Run the dbtran utility to display the starting offset and ending offset of the database transaction log. Make note of the ending offset.

   See “Log Translation utility (dbtran)” *SQL Anywhere Server - Database Administration*.

4. Rename the transaction log to ensure that it is not modified during the unload process. Move the renamed log file to a secure location, such as an offline directory.

5. Unload the database, without using any of the dbunload -a switches to automatically reload the database into another database.

   See “Unload utility (dbunload)” *SQL Anywhere Server - Database Administration*.

6. Initialize a new database.

   See “Initialization utility (dbinit)” *SQL Anywhere Server - Database Administration*.

7. Reload the data into the new database using dbisql to read the reload.sql file generated by dbunload.

   See “Interactive SQL utility (dbisql)” *SQL Anywhere Server - Database Administration*.

8. Shut down the new database.

9. Erase the new database’s transaction log.

10. Run dblog on the new database, using the following options:

    - Use -z to specify the ending offset that you noted earlier.
    - Use -x to set the relative offset to zero.

    For example:

    ```
    dblog -x 0 -z 137829 database-name.db
    ```

   See “Transaction Log utility (dblog)” *SQL Anywhere Server - Database Administration*.

11. Start dbmlsync, specifying the location of the original log file that you moved earlier.
12. When you no longer need the old log file, set the database option delete_old_logs.

See “delete_old_logs option [SQL Remote]” [SQL Anywhere Server - Database Administration].

Results

The remote SQL Anywhere database is unloaded and reloaded.

**UltraLite upgrades**

Consider the following notes prior to upgrading to an UltraLite version 16 database:

- UltraLite version 16 cannot read UltraLite databases that were created using previous versions of the software.

- You cannot upgrade databases on devices.

- Once the database is upgraded, it cannot connect to prior versions of the applications, utilities, and software.

- If you have multiple versions of SQL Anywhere on your computer, pay attention to your system path to ensure that you are using the appropriate utilities. UltraLite utility names are the same between versions, and only the directory path name sets them apart from each other.

Before using existing applications with this version of the software, be sure to review the list of new features and behavior changes to determine whether your application is affected.

**Upgrading version 12 and earlier databases**

UltraLite version 12 and earlier databases must be unloaded into a SQL or XML file using the version of the software that was used to create the database. Use the ulunload utility provided with that software.

If the older version of the software is not installed and you are using a Windows desktop operating system, you can use one of the older ulunload utilities that are included in SQL Anywhere version 16. Select the ulunload utility that is appropriate for your version of the UltraLite database.

Once the database is unloaded, use the version 16 ulload utility to load the XML file into a version 16 database, or the dbisql utility if you unloaded the database to a SQL file.

**Upgrading version 12.0.1 Java edition databases**

Use the UltraLite Java edition version 12.0.1 ULjUnload utility to generate an XML file that can be used by the version 16 uljload utility to create a new UltraLite Java edition database.

**Compatibility with existing software**

- UltraLite 16 database files only support connections from version 16 client applications or the version 16 UltraLite engine.

---

Copyright © 2014, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0 311
The UltraLite version 16 runtime and the UltraLite version 16 engine do not work with database files and application code created with UltraLite version 9 and earlier.

See also
- “UltraLite new features” on page 249
- “UltraLite behavior changes and deprecated features” on page 254
- “How to ensure that you are running the correct version of the utilities when you have multiple versions installed” on page 275

Upgrading an UltraLite database (Windows)

For users of previous versions of the software, this task summarizes the process of rebuilding your UltraLite database for version 16.

Prerequisites

- If you are upgrading from an UltraLite version 9 or earlier database, you must have that version of SQL Anywhere installed.
- Make a backup copy of your existing UltraLite database.
- Synchronize your database if it is a production database that may contain unsynchronized changes.

Context and remarks

UltraLite version 16 cannot read UltraLite databases created using any prior version of the software.

Task

1. Create an XML file (or files) with the contents of the database.

   Open a command prompt, navigate to the directory of the earlier SQL Anywhere install, and then use the ununload utility on the database.

   If the UltraLite database was created with version 10, 11, or 12 of SQL Anywhere, you can, alternatively, open a command prompt, go to the %SQLANY16%\UltraLite\Unload\ directory, choose the directory that applies to your version of the UltraLite database, such as V10, V11, or V12, and then use the ununload utility from that directory.

2. Use the UltraLite version 16 ulload utility, or use the Load Database Wizard in the UltraLite 16 plug-in for Sybase Central to load the schema and data into a new version 16 database.

   UltraLite databases now default to the UTF8 encoding. If this encoding does not suit your needs, you should explicitly set the utf8_encoding parameter to off.

3. Check the generated XML file to verify the setting of the UTF-8 encoding.

Results

The UltraLite database is upgraded to the latest version.
Upgrading an UltraLite database (Linux)

For users of previous versions of the software, this task summarizes the process of rebuilding your database to version 16.

**Prerequisites**

- Install the version of SQL Anywhere that created the database.
- Make a backup copy of your existing UltraLite database.
- Synchronize your database if it is a production database that may contain unsynchronized changes.

**Context and remarks**

UltraLite version 16 cannot read UltraLite databases created using any prior version of the software.

**Task**

1. Create an XML file (or files) with the contents of the database.

   At a command prompt, go to `%SQLANY%/bin32`—where `%SQLANY%` is the location of your earlier SQL Anywhere install—and then use the ulunload utility on the databases.

2. Use the UltraLite version 16 ulload utility, or use the Load Database Wizard in the UltraLite 16 plug-in for Sybase Central to load the schema and data into a new version 16 database.

   UltraLite databases now default to the UTF8 encoding. If this encoding does not suit your needs, you should explicitly set the utf8_encoding parameter to off.

3. Check the generated XML file to verify the setting of the UTF-8 encoding.

**Results**

The UltraLite database is upgraded to the latest version.

See also

- “UltraLite utf8_encoding creation parameter” [UltraLite - Database Management and Reference]
- “UltraLite Database Unload utility (ulunload)” [UltraLite - Database Management and Reference]
- “UltraLite Load XML to Database utility (ulload)” [UltraLite - Database Management and Reference]
SQL Remote upgrades

SQL Remote installations include a consolidated database and many remote databases, together with a SQL Remote Message Agent at each site.

At each site, the SQL Remote Message Agent handles the sending and receiving of messages. The messages take the form of SQL statements, and the database server handles the actual execution of those SQL statements.

The upgrade requirements for SQL Remote are as follows:

- **Software upgrades can be one site at a time**  
  Older Message Agents (dbremote) can exchange messages with version 16 Message Agents. For version 5 of SQL Remote, the version 5 Message Agents can exchange messages with version 16 Message Agents, as long as the compression database option is set to a value of -1. There is no need to upgrade software throughout the installation simultaneously. See “compression option [SQL Remote]” [SQL Anywhere Server - Database Administration].

- **Upgrade databases**  
  If you are upgrading a remote or consolidated database that used SQL Anywhere version 9 or lower, you must upgrade the database file format by unloading and reloading your database. There is no need for all databases to be upgraded at the same time.

  For instructions on unloading and reloading the database, see “The rebuild process for version 9 and earlier databases” on page 280.

- **Upgrading Adaptive Server Enterprise consolidated databases**  
  SQL Remote no longer supports Adaptive Server Enterprise consolidated databases. To synchronize Adaptive Server Enterprise databases, you should upgrade to MobiLink.

  For information about migrating from SQL Remote to MobiLink, see http://www.sybase.com/detail?id=1034174.

- **Upgrading with SQL Remote**  
  Support for the VIM and MAPI message systems for SQL Remote was removed in version 11.0.0. When you upgrade a database that uses VIM or MAPI to SQL Anywhere version 16, you must change the message type to File, FTP, or SMTP. If the message type is MAPI or VIM, dbremote.exe does not start.

**Example**

One approach to upgrading version 5 of SQL Remote is as follows:

1. Upgrade the consolidated database server and SQL Remote Message Agent, and then upgrade the database file by unloading and reloading the consolidated database. Set the compression database option to -1, so that all messages are compatible with the version 5 software at remote sites. To unload and reload the consolidated database, see “Rebuilding databases involved in synchronization or replication (command line)” [SQL Anywhere Server - SQL Usage].

2. One at a time, upgrade the remote database servers and Message Agents, and then upgrade the database file format by unloading and reloading the remote databases. You can set the compression database option to a value other than -1 to take advantage of compression and encoding on messages.
being sent to the consolidated database server. To unload and reload the remote databases, see “Rebuilding databases involved in synchronization or replication (command line)” [SQL Anywhere Server - SQL Usage].

3. When all remote database servers and Message Agents are upgraded, set the compression database option at the consolidated site to a value other than -1.

## Upgrading the SQL Anywhere Monitor and migrating resources

Use the Migrator utility to migrate the resources from one Monitor to a newly installed Monitor.

### Prerequisites

**Caution**
Uninstalling the Monitor removes the application, as well as resources and collected metrics. To preserve your current Monitor resources, you must:

1. Install a new version of the Monitor.
2. Migrate the resources.
3. Uninstall the older version of the Monitor.

To avoid unwanted alerts, schedule a blackout period in the SQL Anywhere Monitor for your servers before you shut them down.

### Task

1. Create a back up copy of the existing Monitor database file, `samonitor.db`.

   The default locations of the version 11.0.1 Monitor and version 16.0 Monitor database files are listed in the following table:

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Monitor type</th>
<th>version 11.0.1 and 12 directories</th>
<th>version 16.0 directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows XP</td>
<td>Monitor Devel-</td>
<td>C:\Documents and</td>
<td>C:\Documents and</td>
</tr>
<tr>
<td></td>
<td>oper Edition</td>
<td>Settings\All Users</td>
<td>Settings\All Users</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\Documents\SQL Anywhere</td>
<td>\Documents\SQL Anywhere</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11\Monitor\samonitor.db</td>
<td>16\Monitor\samonitor.db</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating system</td>
<td>Monitor type</td>
<td>version 11.0.1 and 12 directories</td>
<td>version 16.0 directory</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Windows XP</td>
<td>Monitor Production Edition</td>
<td>C:\Documents and Settings\All Users \Documents\SQL Anywhere 11 Monitor\samonitor.db</td>
<td>C:\Documents and Settings\All Users \Documents\SQL Anywhere 16 Monitor \samonitor.db</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:\Documents and Settings\All Users \Documents\SQL Anywhere 12 Monitor\samonitor.db</td>
<td></td>
</tr>
<tr>
<td>Windows Vista and later versions of Windows</td>
<td>Monitor Developer Edition</td>
<td>C:\Users\Public \Documents\SQL Anywhere 11 Monitor\samonitor.db</td>
<td>C:\Users\Public \Documents\SQL Anywhere 16 Monitor \samonitor.db</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:\Users\Public \Documents\SQL Anywhere 12 Monitor\samonitor.db</td>
<td></td>
</tr>
<tr>
<td>Windows Vista and later versions of Windows</td>
<td>Monitor Production Edition</td>
<td>C:\Users\Public \Documents\SQL Anywhere 11 Monitor\samonitor.db</td>
<td>C:\Users\Public \Documents\SQL Anywhere 16 Monitor \samonitor.db</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C:\Users\Public \Documents\SQL Anywhere 12 Monitor\samonitor.db</td>
<td></td>
</tr>
<tr>
<td>Linux</td>
<td>Monitor Developer Edition</td>
<td>/opt/sqlanywhere11/ samonitor.db</td>
<td>/opt/sqlanywhere16/ samonitor.db</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/opt/sqlanywhere12/ samonitor.db</td>
<td></td>
</tr>
<tr>
<td>Linux</td>
<td>Monitor Production Edition</td>
<td>/opt/samonitor11/ samonitor.db</td>
<td>/opt/samonitor12/ samonitor.db</td>
</tr>
</tbody>
</table>

2. Install the new Monitor. Run the `setup.exe` file from the `Monitor` directory on your installation media, and follow the instructions provided. When the installation finishes, stop the new Monitor (if it is running).

Only one version of the Monitor can run on a computer at a time.

When you install the version 16.0 Monitor on a computer where an older version of the Monitor is running, the install stops the old version of the Monitor.

3. At a command prompt, run the Migrator utility.

Run the Migrator utility with the following options:

- `-t temporary-directory` Specifies the directory for temporary files. By default, the temporary files are created in the same directory as the `run_migrator` file.
**Note**
The Monitor Migrator creates temporary files that are deleted at the end of the migration process. Use the -t option to specify a directory for these temporary files. The temporary files take up a similar amount of space as the old Monitor database file. Ensure that the specified directory has sufficient space.

- **source-filename**  Specifies the path and file name to the old Monitor database file. For example, the path to the version 11.0.1 `samonitor.db` file.

- **destination-filename**  Specifies the path and file name to the new Monitor file where the resources and configuration settings are loaded. For example, the path to the version 16.0 `samonitor.db` file.

For example:

```
C:\Program Files\SQL Anywhere 16\run_migrator.cmd -t c:\monitorbackup c:\Program Files\SQL Anywhere 11\Monitor\samonitor11.db C:\Program Files\SQL Anywhere 16\Monitor\samonitor16.db
```

Use the following table to determine the location of the Migrator utility.

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Monitor type</th>
<th>Default location of the Migrator utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>Monitor Developer Edition</td>
<td>C:\Program Files\SQL Anywhere 16\Bin32\run_migrator.cmd</td>
</tr>
<tr>
<td></td>
<td>Monitor Production Edition</td>
<td>C:\Program Files\SQL Anywhere 16\Bin32\run_migrator.cmd</td>
</tr>
<tr>
<td>Linux</td>
<td>Monitor Developer Edition</td>
<td>/opt/sqlanywhere16/bin32/run_migrator.sh</td>
</tr>
<tr>
<td></td>
<td>Monitor Production Edition</td>
<td>/opt/sqlanywhere16/bin32/run_migrator.sh</td>
</tr>
</tbody>
</table>

**Results**
The Monitor is upgraded.

**Next**
Start the Monitor

**See also**
- “Starting the Monitor” [SQL Anywhere Server - Database Administration]
Index

Symbols

#hook_dict table
  version 16.0 behavior change, MobiLink, 52
% operator
  version 11.0.0 behavior change, 241
-ac option
  version 12.0.0 enhancement, dbsupport utility, 124
-al option
  version 16.0 new feature, 23
-aw option
  version 16.0 new feature, 19
-bc option
  version 16.0 new feature, 19
-bk option
  version 12.0.0 new feature, MobiLink, 154
-bkr option
  version 12.0.0 new feature, MobiLink, 154
-c option
  version 11.0.1 enhancement, MobiLink (mlmon), 198
  version 12.0.1 behavior change, 92
-ca option
  version 12.0.0 new feature, MobiLink, 151
-cc option
  version 16.0 behavior change, 35
-ch option
  version 16.0 dbsupport unsupported, 36
  version 16.0 enhancement, 33
-chx option
  version 12.0.0 new feature, 139
-ci option
  version 12.0.0 new feature, MobiLink, 154
-cin1 option
  version 12.0.0 new feature, MobiLink, 151
  version 12.0.1 enhancement, MobiLink, 95
-cl option
  version 12.0.0 new feature, MobiLink, 155
  version 12.0.1 behavior change, 92
-cm option
  version 11.0.1 new feature, dbunload, 188
  version 12.0.0 new feature, MobiLink, 155
  version 12.0.1 behavior change, MobiLink (mlsrv12), 99
  version 16.0 unsupported, 41
-cmax option
  version 12.0.0 new feature, MobiLink, 151
  version 12.0.1 enhancement, MobiLink, 95
-cmin option
  version 12.0.0 new feature, MobiLink, 151
  version 12.0.1 enhancement, MobiLink, 95
-cn option
  version 12.0.0 behavior change, MobiLink, 157
-codepage option
  version 11.0.0 deprecated, 263
  version 12.0.0 deprecated, 182
-cp option
  version 11.0.0 enhancement, dbunload, 212
  version 16.0 behavior change, 35
-cs option, MobiLink (mlsrv11)
  version 11.0.0 new feature, 246
-cw option
  version 16.0 unsupported, 41
df option
  version 12.0.0 behavior change, MobiLink, 158
  version 12.0.0 enhancement, MobiLink, 158
dp option
  version 12.0.0 behavior change, MobiLink, 159
  version 12.0.0 enhancement, MobiLink, 158
e option
  version 11.0.0 unsupported, initialization utility (dbinit) option, 241
db database server option
  version 16.0 deprecated, 13
es option
  version 11.0.0 new feature, 213
-f option
  version 12.0.0 discontinued, MobiLink, 160
-fr option
  version 12.0.0 discontinued, MobiLink, 160
-ftru option
  version 12.0.0 new feature, MobiLink, 151
g option
  version 11.0.0 deprecated, MobiLink Listener utility (dblsn) option, 248
  version 11.0.0 enhancement, dbunload, 212
  version 12.0.0 removed, transaction log utility (dblog) option, 114
ga option
  version 11.0.0 deprecated, MobiLink Listener utility (dblsn) option, 248
gb option
  version 11.0.0 enhancement, 213
Index

-gd option
  version 16.0 behavior change, 9
-gi option
  version 11.0.0 deprecated, MobiLink Listener utility (dbsln) option, 248
-gk option
  version 16.0 behavior change, 8
-gl option
  version 16.0 behavior change, 9
-gn option
  version 12.0.0 enhancement, 120
-gna option
  version 12.0.0 enhancement, 120
  version 12.0.1 behavior change, 92
-gnh option
  version 12.0.0 enhancement, 120
  version 12.0.1 behavior change, 92
-gnl option
  version 12.0.0 enhancement, 120
  version 12.0.1 behavior change, 92
-gns option
  version 12.0.0 new feature, 120
  version 12.0.1 behavior change, 92
-gss option
  version 11.0.0 behavior change, 235
  version 11.0.0 enhancement, 227
-gta option
  version 16.0 new feature, 23
-gu option
  version 12.0.0 behavior change, 147
  version 16.0 behavior change, 8
-gx option
  version 11.0.0 unsupported, 235
-h option
  version 16.0 new feature, 19
-i option
  version 12.0.0 new feature, MobiLink, 156
-il option
  version 12.0.0 removed, transaction log utility (dblog) option, 114
-im option
  version 11.0.0 new feature, 213
-is option
  version 12.0.0 behavior change, log translation utility (dbtran) option, 113
-k option
  version 12.0.0 enhancement, dblic utility, 124
-kd option
  version 12.0.0 enhancement, dbunload utility, 124
-kp option
  version 12.0.1 new feature, 84
-kr option
  version 12.0.1 deprecated, 93
-ks option
  version 11.0.0 new feature, 214
-ksc option
  version 11.0.0 new feature, 214
-ksd option
  version 11.0.0 new feature, 214
-l option
  version 11.0.1 new feature, dbunload, 188
-ls option
  version 12.0.1 new feature, MobiLink (mlreplay), 96
  version 16.0 new feature, MobiLink, 51
-lsc option, MobiLink (mlsrv11)
  version 11.0.0 new feature, 246
-lu option
  version 16.0 new feature, MobiLink, 52
-n option
  version 12.0.0 discontinued, MobiLink, 160
-ni option
  version 11.0.0 new feature, dbsln, 247
-no option
  version 11.0.0 enhancement, dbunload, 212
-ns option
  version 16.0 behavior change, 51
-nu option
  version 11.0.0 new feature, dbsln, 248
  version 16.0 behavior change, 51
-o option
  version 12.0.1 behavior change, MobiLink (mlgenreplayapi), 99
  version 12.0.1 behavior change, MobiLink (mlreplay), 99
-os option
  version 12.0.1 new feature, MobiLink (mlgenreplayapi), 96
  version 12.0.1 new feature, MobiLink (mlreplay), 96
-ot option
  version 12.0.1 behavior change, MobiLink (mlgenreplayapi), 99
  version 12.0.1 behavior change, MobiLink (mlreplay), 99
-p option
  version 16.0 enhancement, 33
-pc option
  version 11.0.0 new feature, dblsn, 247
-pd option
  dbinit, version 16.0 new feature, 32
dbupgrad, version 16.0 new feature, 33
-ppv option
  version 11.0.1 enhancement, MobiLink (mlsrv11), 197
  version 12.0.0 enhancement, MobiLink, 151
-qi option
  version 12.0.1 behavior change, MobiLink, 82
-qr option
  version 12.0.0 enhancement, dbunload utility, 124
-r option
  version 12.0.0 discontinued, MobiLink, 160
-rep option
  version 12.0.1 new feature, MobiLink (mlreplay), 96
-rnt option
  version 12.0.1 new feature, MobiLink (mlreplay), 96
-rp option
  version 12.0.0 new feature, MobiLink, 151
version 12.0.1 new feature, MobiLink (mlreplay), 96
-rpp option
  version 12.0.0 new feature, MobiLink, 151
-rs option
  version 12.0.0 enhancement, dbsvc utility, 124
-rsu option
  version 12.0.0 removed, log translation utility (dbtran) option, 113
-s option
  version 12.0.0 enhancement, initialization utility (dbinit), 140
version 12.0.0 new feature, MobiLink, 153, 154
-sbx option
  version 16.0 new feature, 21
-sci option
  version 12.0.1 behavior change, MobiLink (mlreplay), 99
  version 16.0 behavior change, MobiLink (mlreplay), 45
-sf option
  version 11.0.0 enhancement, 213, 227
version 16.0 enhancement, 19
-sk option
  version 16.0 enhancement, 19
-sl dnt
  version 12.0.0 behavior change, MobiLink, 157
-sl java option
  version 12.0.0 behavior change, MobiLink, 151
-sm option
  version 11.0.0 new feature, 209
  version 11.0.1 behavior change, MobiLink, 198
version 12.0.0 behavior change, 147
-sp option
  version 11.0.0 new feature, dbmlsync, 247
-ss option
  version 12.0.0 behavior change, MobiLink, 161
-sv option
  version 11.0.1 enhancement, MobiLink (dblsn), 197
  version 16.0 new feature, MobiLink (mlreplay), 45
-t option
  version 12.0.0 behavior change, service utility (dbsvc) option, 113
-tc option, MobiLink (mlsrv11)
  version 11.0.0 new feature, 246
-tdsl option
  version 16.0 new feature, 19
-tf option, MobiLink (mlsrv11)
  version 11.0.0 new feature, 246
-ts option
  version 16.0 new feature, MobiLink dblsn option, 43
version 16.0 new feature, MobiLink dbmlsync option, 43
version 16.0 new feature, MobiLink mlsrv16 option, 43
-u option
  version 12.0.0 deprecated, dbmlsync, 161
-uc server option
  version 11.0.0 behavior change, 240
-ud option
  version 12.0.0 behavior change, 114
-uf option
  version 16.0 enhancement, 23
-ufd option

Copyright © 2014, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0 321
<table>
<thead>
<tr>
<th>Feature</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessibility</td>
<td>12.0.0</td>
<td>Enhancement, 176</td>
</tr>
<tr>
<td>active_cookie option</td>
<td>12.0.0</td>
<td>New feature, Relay Server, 156</td>
</tr>
<tr>
<td>active_header</td>
<td>12.0.0</td>
<td>New feature, Relay Server, 156</td>
</tr>
<tr>
<td>Adaptive Server Enterprise</td>
<td>12.0.1</td>
<td>New feature, MobiLink, 95</td>
</tr>
<tr>
<td>Adaptive Server Enterprise 12.5.x</td>
<td>12.0.0</td>
<td>Discontinued, MobiLink, 160</td>
</tr>
<tr>
<td>Adaptive Server Enterprise support</td>
<td>12.0.1</td>
<td>New feature, MobiLink, 94</td>
</tr>
<tr>
<td>addBatch</td>
<td>12.0.0</td>
<td>Enhancement, 135</td>
</tr>
<tr>
<td>address space</td>
<td>12.0.0</td>
<td>Behavior change, 141</td>
</tr>
<tr>
<td>Address Windowing Extensions</td>
<td>12.0.0</td>
<td>Deprecated, 146</td>
</tr>
<tr>
<td>administration tools</td>
<td>11.0.1</td>
<td>Behavior change, 196</td>
</tr>
<tr>
<td>ADO.NET</td>
<td>11.0.1</td>
<td>Enhancement, 191</td>
</tr>
<tr>
<td>version 12.0.1 enhancement, 133</td>
<td></td>
<td></td>
</tr>
<tr>
<td>version 16.0 enhancement, 66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADSODBC server class</td>
<td>11.0.0</td>
<td>New feature, 230</td>
</tr>
<tr>
<td>AES 256-bit encryption</td>
<td>16.0</td>
<td>New feature, UltraLite, 59</td>
</tr>
<tr>
<td>AES encryption algorithm</td>
<td>16.0</td>
<td>Enhancement, 30</td>
</tr>
<tr>
<td>AES256 encryption algorithm</td>
<td>11.0.0</td>
<td>New feature, 210</td>
</tr>
<tr>
<td>AES256_FIPS encryption algorithm</td>
<td>11.0.0</td>
<td>New feature, 210</td>
</tr>
<tr>
<td>affinity flag extended</td>
<td>16.0</td>
<td>Enhancement, Relay Server, 3</td>
</tr>
<tr>
<td>aggregate functions</td>
<td></td>
<td>Outer references, 298</td>
</tr>
<tr>
<td>alerts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitor version 12.0.1 new feature, 110, 111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>aliases</td>
<td>16.0</td>
<td>Enhancement, 31</td>
</tr>
<tr>
<td>ALLOW clause</td>
<td>16.0</td>
<td>New feature, 37</td>
</tr>
<tr>
<td>allow_read_client_file property</td>
<td>11.0.0</td>
<td>New feature, 212</td>
</tr>
<tr>
<td>allow_write_client_file option</td>
<td>11.0.0</td>
<td>New feature, 213</td>
</tr>
<tr>
<td>allow_write_client_file property</td>
<td>11.0.0</td>
<td>New feature, 214</td>
</tr>
<tr>
<td>allowPasswordsInFavorites</td>
<td>12.0.0</td>
<td>New feature, 180</td>
</tr>
<tr>
<td>ALTER DATABASE statement</td>
<td>11.0.0</td>
<td>Enhancement, 221</td>
</tr>
<tr>
<td>version 16.0 enhancement, 26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>version 16.0 new feature, 17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALTER EXTERNAL ENVIRONMENT statement</td>
<td>12.0.0</td>
<td>Enhancement, 139</td>
</tr>
<tr>
<td>ALTER FUNCTION statement</td>
<td>11.0.0</td>
<td>Enhancement, 219</td>
</tr>
<tr>
<td>ALTER LDAP SERVER statement</td>
<td>16.0</td>
<td>New feature, 12</td>
</tr>
<tr>
<td>ALTER LOGIN POLICY statement</td>
<td>11.0.0</td>
<td>New feature, 220</td>
</tr>
<tr>
<td>version 16.0 enhancement, 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALTER MATERIALIZED VIEW statement</td>
<td>11.0.0</td>
<td>Enhancement, 219</td>
</tr>
<tr>
<td>ALTER MIRROR SERVER statement</td>
<td>12.0.0</td>
<td>New feature, 118</td>
</tr>
<tr>
<td>ALTER PROCEDURE statement</td>
<td>11.0.0</td>
<td>Enhancement, 219</td>
</tr>
<tr>
<td>version 12.0.0 behavior change, 114</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALTER ROLE statement</td>
<td>16.0</td>
<td>New feature, 9</td>
</tr>
<tr>
<td>ALTER SEQUENCE statement</td>
<td>12.0.0</td>
<td>New feature, 119</td>
</tr>
<tr>
<td>ALTER SERVER statement</td>
<td>12.0.0</td>
<td>Enhancement, 130</td>
</tr>
<tr>
<td>version 12.0.1 behavior change, 76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>version 16.0 behavior change, 39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALTER SPATIAL REFERENCE SYSTEM statement</td>
<td>12.0.0</td>
<td>New feature, 115</td>
</tr>
<tr>
<td>ALTER SYNCHRONIZATION SUBSCRIPTION statement</td>
<td>12.0.0</td>
<td>New feature, 37</td>
</tr>
</tbody>
</table>
version 12.0.0 enhancement, 150
version 12.0.0 enhancement, MobiLink, 154
version 12.0.1 enhancement, OnDemand, 74
version 16.0 enhancement, 48
ALTER SYNCHRONIZATION USER statement
version 12.0.1 enhancement, OnDemand, 74
version 16.0 enhancement, 48
ALTER TABLE statement
version 12.0.0 behavior change, 114
version 12.0.1 enhancement, 77
version 16.0 enhancement, 26, 30
ALTER TEXT CONFIGURATION statement
version 11.0.0 new feature, 220
version 12.0.0 enhancement, 130
ALTER TEXT INDEX statement
version 11.0.0 new feature, 221
ALTER TRACE EVENT SESSION statement
version 16.0 new feature, 27
ALTER USER statement
version 11.0.0 new feature, 220
version 16.0 enhancement, 12
alternate server names
version 12.0.0 behavior change, 142
AlternateMirrorServerName property
version 11.0.0 new feature, 216
an_unload_db structure
version 12.0.0 enhancement, 134
Android
version 12.0.1 enhancement, UltraLite, 104
version 12.0.1 new feature, UltraLite support on Android, 103
version 12.0.1 new support, UltraLite, 101
ansi_integer_overflow option
version 11.0.0 unsupported, 241
ansi_substring option
version 11.0.0 unsupported, 241
version 11.0.1 no longer deprecated, 197
Apache
version 12.0.1 quick deployment, Relay Server, 83
version 16.0 new feature, quick setup for Relay Server, 54
version 16.0, 32-bit Relay Server discontinued, 57
APP connection parameter
version 11.0.0 enhancement, 209
ApplInfo connection parameter
version 11.0.0 enhancement, 209
apply expressions
version 11.0.0 new feature, 222
ApproximateCPUTime property
version 12.0.1 new feature, 90
arbiter servers
version 12.0.1 enhancement, 75
archive backups
version 11.0.1 enhancement, 193
version 12.0.0 enhancement, 123
ARRAY constructor
version 16.0 new feature, 15
array operators
version 16.0 new feature, 15
ARRAY_AGG function
version 16.0 new feature, 15
ARRAY_MAX_CARDINALITY function
version 16.0 new feature, 15
ASE 15.7 support
version 16.0 new feature, MobiLink, 44
ASEJDBC server class
version 12.0.0 deprecated, 146
version 16.0 unsupported, 41
ASP.NET
version 11.0.1 new feature, 191
assertions
version 16.0 behavior change, 34
asterisks
version 11.0.1 behavior change, full text search, 194
asynchronous synchronization
UltraLite.NET, 103
auditing
version 11.0.0 enhancement, 210
version 11.0.0 new feature, 257
authenticate_file_upload connection event
version 12.0.0 new feature, MobiLink, 152
Authenticated property
version 11.0.1 new feature, connection property, 187
version 11.0.1 new feature, database property, 188
authentication parameters
version 12.0.1 new feature, MobiLink, 95
authentication synchronization scripts
version 16.0 enhancement, UltraLite, 60
authorities
version 16.0 behavior change, replaced by roles, 40
AuthType property
version 11.0.0 new feature, 214
auto start option
Relay Server, version 16 unsupported, 57
auto_commit option
  version 12.0.0 behavior change, 182
auto_commit_on_create_local_temp_index option
  version 16.0 behavior change, 33
  version 16.0 new feature, 22
auto_commit_on_create_local_temp_index property
  version 16.0 new feature, 24
auto_refetch
  version 12.0.0 behavior change, 182
auto_unlock_time option
  version 16.0 new feature, 7
autocommit
  version 11.0.0 behavior change, 240
AUTOINCREMENT
  retaining the next available value in the rebuilt database, 277
  using the reset.sql script to retain the next available value, 277
automatic cache sizing
  version 12.0.1 new feature, UltraLite, 102
automatic connection pooling
  version 12.0.0 enhancement, 134
automatic_timestamp option
  version 11.0.0 unsupported, 241
AutoMultiProgrammingLevel property
  version 12.0.0 new feature, 127
AutoMultiProgrammingLevelStatistics property
  version 12.0.0 new feature, 127
AWE
  version 12.0.0 deprecated, 146
  version 16.0 unsupported, 41
backup database wizard
  version 12.0.0 enhancement, 178
BACKUP statement
  version 11.0.1 enhancement, 193
  version 12.0.0 enhancement, 123
backup utility (dbbackup)
  version 16.0 enhancement, 19
BackupInProgress
  version 16.0 enhancement, 2
backups
  version 11.0.1 enhancement, 193
  version 12.0.0 enhancement, 123
batches
  version 11.0.0 enhancement, 229
BatchUpdateCount
  version 12.0.0 enhancement, 135
BatchUpdateException
  version 12.0.0 enhancement, 135
BEGIN SNAPSHOT statement
  version 11.0.0 new feature, 221
BEGIN statement
  version 12.0.0 enhancement, 130
behavior changes
  version 11.0.0, 205
  version 11.0.1, 187
  version 12.0.0, 113
  version 12.0.1, 73
  version 16.0, 1
  version 16.0, SQL Remote, 72
  version 9.0.2 and earlier, vii
bell option
  version 12.0.0 behavior change, 182
big-endian UTF-16 encoding
  version 11.0.0 new feature, 228
BIGINT data type
  version 12.0.0 behavior change, MobiLink, 158
BIGINT UNSIGNED data type
  version 16.0 new feature, 28
bin32 directory
  version 11.0.0 behavior change, 266
bin64 directory
  version 11.0.0 behavior change, 266
BINSEARCH protocol option
  version 11.0.0 unsupported, 241
BINTOHEX function
  version 16.0 new feature, 25
BIT_AND function
  version 12.0.0 enhancement, 128

B

back quotes
  version 12.0.0 new feature, 139
Background synchronization
  version 11.0.0 enhancement, UltraLite, 252
background synchronization
  version 12.0.0 enhancement, MobiLink, 153
background_priority option
  version 11.0.0 deprecated, 243
backup and recovery
  version 11.0.0 enhancement, 210
  version 12.0.0 enhancement, 123
  version 16.0 enhancement, 19
BACKUP authority
  version 16.0 deprecated, 40
Index

BIT_OR function
version 12.0.0 enhancement, 128

BIT_XOR function
version 12.0.0 enhancement, 128

BlackBerry
version 11.0.0 new feature, UltraLite support on BlackBerry, 254

BlackBerry Enterprise Server
version 12.0.1 enhancement, 81

BLOBs
version 11.0.0 new feature, querying, 209
blocking
version 11.0.0 behavior change, 239
version 12.0.0 enhancement, 124
version 16.0 enhancement, MobiLink, 43
blocking_others_timeout option
version 12.0.0 new feature, 124
broadcast repeater utility (dbns12)
version 12.0.0 behavior change, 114
BufferDownload extended option
version 12.0.0 new feature, MobiLink, 155
bypass queries
version 12.0.1 enhancement, 89

BYTE ORDER MARK
version 11.0.0 enhancement, 262

BytesReceived property
version 12.0.1 new feature, 90
BytesReceivedUncomp property
version 12.0.1 new feature, 90
BytesSent property
version 12.0.1 new feature, 90
BytesSentUncomp property
version 12.0.1 new feature, 90

C

CacheRead property
version 11.0.0 new feature, 239
CacheReadEng property
version 11.0.0 renamed, 239
CacheReadWorkTable property
version 11.0.0 new feature, 214, 216

CALL statement
version 12.0.0 deprecated, calling functions, 146
callbacks in C API
version 16.0 enhancement, 2

CancelSync method
version 12.0.0 new feature, MobiLink, 155

CARDINALITY function
version 16.0 new feature, 15

CarverHeapPages property
version 12.0.1 new feature, 90

CASE expression
version 11.0.0 enhancement, 221
version 11.0.0 enhancement, UltraLite, 253

CASE statement
version 11.0.0 enhancement, 221
version 11.0.0 enhancement, UltraLite, 253
version 12.0.0 enhancement, 131
catalog
version 11.0.0 behavior change, 232

CATALOGS rowset, OLE DB
version 11.0.1 new feature, 191
central administration of remote databases
version 12.0.0 enhancement, Relay Server, 156
version 12.0.0 new feature, MobiLink, 148
version 12.0.1 enhancements, MobiLink, 95
version 12.0.1 new feature, MobiLink, 97

Certicom FIPS
version 16.0 unsupported feature, 3
certificate creation utility (viewcert)
version 16.0 enhancement, 22
certificate protocol option
version 11.0.0 behavior change, 248
version 11.0.0 unsupported, 266
certificate viewer utility (viewcert)
version 16.0 enhancement, 22
certificate_password protocol option
version 11.0.0 behavior change, 248
version 11.0.0 unsupported, 266
certificates
version 11.0.0 behavior change, 266
version 16.0 behavior change, 12
version 16.0 behavior change for FIPS, 3

CESU-8 character set
version 16.0 enhancement, 31
CESUSBIN collation
  version 16.0 enhancement, 31
change_password_dual_control option
  version 16.0 new feature, 18
changes in version
  11.0.0, 205
  11.0.1, 187
  12.0.0, 113
  12.0.1, 73
  16.0, 1
  version 9.0.2 and earlier, vii
CHAR data type
  version 12.0.0 enhancement, 133
character data types
  version 12.0.0 enhancement, 133
check for updates
  version 12.0.0 enhancement, 176
checkpoint logs
  version 12.0.0 behavior change, 140
checkpoint method
  version 16.0 removed API, UltraLite, 65
checkpointing
  tables with autoincrement columns, 277
  version 12.0.0 behavior change, 140
CheckpointLogBitmapPagesWritten property
  version 12.0.0 unsupported, 146
CheckpointLogBitmapSize property
  version 12.0.0 unsupported, 146
checkpoints
  version 12.0.0 behavior change, 140
CHECKSUM clause
  version 12.0.0 enhancement, 133
  version 12.0.0 enhancement, ALTER DATABASE
  statement, 122
  version 12.0.0 enhancement, CREATE DATABASE statement, 140
  version 12.0.0 enhancement, START DATABASE
  statement, 122
checksums
  version 11.0.0 behavior change, 236
  version 11.0.0 enhancement, 227
  version 12.0.0 behavior change, 140
  version 12.0.0 enhancement, 122
chunk mode transfer-coding for HTTP clients
  version 11.0.1 behavior change, 197
ClassLoader
  version 12.0.1 enhancement, 88
CLEAR statement
  version 12.0.0 behavior change, 182
clearBatch
  version 12.0.0 enhancement, 135
callback authentication using common access cards
  version 11.0.1 new feature, 198
client files, reading from
  version 11.0.0 new feature, 207
client files, writing to
  version 11.0.0 new feature, 207
client libraries
  version 16.0 behavior change, 38
ClientNodeAddress property
  version 11.0.0 new feature, 214
ClientStmtCacheHits property
  version 12.0.1 new feature, 90
ClientStmtCacheMisses property
  version 12.0.1 new feature, 90
cloud support
  version 12.0.1 new feature, 73
code sample locations
  UltraLiteJ, 103
collation tailoring support in the SQL Anywhere plug-in
  version 11.0.0 new feature, 257
collations
  version 12.0.0 behavior change, 145
collect_network_data protocol option
  version 12.0.1 new feature, MobiLink, 82
column attributes
  retaining the next available value in the rebuilt database, 277
column mapping
  version 16.0 behavior change, 50
column names
  version 16.0 behavior change, 14
  version 16.0 behavior change, UltraLite, 66
columns
  version 16.0 enhancement, 30
command line utilities
  multiple versions, 275
  UltraLite multiple versions, 311
  UltraLite upgrading, 311
  upgrading, 275
CommBufferSize connection parameter
  version 16.0 enhancement, 33
COMMENT statement
  version 11.0.0 enhancement, 221
  version 11.0.1 deprecated clause, 197
version 12.0.0 enhancement, 118, 119
version 16.0 enhancement, 12
Commit property
version 12.0.1 new feature, 91
COMMIT statement
version 12.0.0 enhancement, 180
commit_on_exit option
version 12.0.0 behavior change, 182
common access cards
version 11.0.1 new feature, 198
Compact Framework 3.5 support
version 16.0 new feature, 28
compare databases window
version 12.0.1 new feature, 75
compatibility
client/server, 272
databases and database servers, 272
issues, 272
UltraLite software upgrades, 311
composite data types
version 16.0 new feature, 15
compressed columns
version 11.0.0 enhancement, 229
computed columns
version 11.0.0 behavior change, 235
ConfigPersistent.setRowScoreFlushSize method
[UltraLiteJ]
version 12.0.1 enhancement, 81
ConfigPersistent.setRowScoreMaximum method
[UltraLiteJ]
version 12.0.1 enhancement, 81
configuration files
version 11.0.0 enhancement, 212
version 12.0.0 enhancement, 113, 124
conflict detection
version 16.0 removed, MobiLink, 53
conflict handling
version 16.0 enhancement, MobiLink plug-in, 49
conflicted_deletes property
version 16.0 removed, MobiLink, 53
conflicted Inserts property
version 16.0 removed, MobiLink, 53
connect assistant
version 11.0.0 new feature, 256
version 12.0.0 removed, 175
Connect method
version 16.0 new feature, Dbmlsync .NET API, 47
version 16.0 new feature, Dbmlsync C++ API, 47
Connect window
version 12.0.1 enhancement, OnDemand, 73
version 16.0 behavior change, 67
connection pool
version 12.0.1 enhancement, 12
version 11.0.0 enhancement, 256
version 11.0.0 new feature, 256
version 12.0.0 enhancement, 175
version 12.0.0 new feature, 108
ConnectedTime property
version 12.0.1 new feature, 89
CONNECTION CLOSE clause
version 12.0.1 behavior change, 76
version 16.0 unsupported, 39
connection pooling
version 12.0.0 enhancement, 134
version 12.0.0 new feature, 122
connection properties
version 11.0.0 new feature, 214
version 12.0.0 new feature, 126
version 12.0.1 enhancement, 76
version 12.0.1 new feature, 89
version 16.0 enhancements, 24
connection states on Android
version 16.0 enhancement, UltraLiteJ, 63
Connection.getConnection method [UltraLite for M-Business Anywhere]
version 12.0.1 new feature, 81
CONNECTION_EXTENDED_PROPERTY function
version 16.0 behavior change, 36
CONNECTION_PROPERTY function
version 16.0 behavior change, 36
ConnectionPool connection parameter
version 12.0.0 new feature, 122
connections
version 11.0.0 behavior change, 236
ConnPoolCachedCount property
version 12.0.0 new feature, 126
ConnPoolHits property
version 12.0.0 new feature, 126
ConnPoolMisses property
version 12.0.0 new feature, 126
console utility (dbconsole)
version 16.0 new feature, 71
consolidated database support
version 12.0.0 new feature, MobiLink, 150
consolidated databases
IBM DB2 LUW, 303
upgrading, 300
upgrading Adaptive Server Enterprise, SAP Sybase IQ, Oracle, MySQL, or Microsoft SQL Server, 302
upgrading SQL Anywhere 10.0.0 and later, 301
upgrading SQL Anywhere versions earlier than version 10.0.0, 305
CONTAINS search condition
version 11.0.0 new feature, 220
version 16.0 enhancement, 31
CONVERT function
version 12.0.1 behavior change, 93
cookies
Relay Server, 55
cooperaive_commit_timeout option
version 11.0.0 behavior change, 237
cooperaive_commits option
version 11.0.0 behavior change, 237
copy connection string to clipboard
version 11.0.0 new feature, 256
copy nodes
version 12.0.0 new feature, 118
CopyNodeParent
version 16.0 new feature, 17
correlation names
version 12.0.0 enhancement, 133
corrupt databases
version 12.0.0 enhancement, 140
COUNT_BIG function
version 12.0.0 new feature, 129
Counter1 statistic
version 12.0.1 new feature, 91
Counter2 statistic
version 12.0.1 new feature, 91
Counter3 statistic
version 12.0.1 new feature, 91
Counter4 statistic
version 12.0.1 new feature, 91
Counter5 statistic
version 12.0.1 new feature, 91
CPOOL connection parameter
version 12.0.0 new feature, 122
CPUs
version 16.0 behavior change, 35
CRC32 algorithm
version 12.0.0 enhancement, 128
CREATE CERTIFICATE statement
version 16.0 new feature, 26
CREATE DATABASE statement
version 16.0 behavior change, 35
version 16.0 enhancement, 28, 31, 32
create database wizard
version 11.0.0 enhancement, 260
version 12.0.0 enhancement, 179
CREATE DECRIPTED DATABASE statement
version 11.0.1 new feature, 189
version 16.0 behavior change, 35
version 16.0 enhancement, 28
CREATE DECRYPTED FILE statement
version 16.0 behavior change, 35
version 16.0 enhancement, 28
CREATE ENCRYPTED DATABASE statement
version 11.0.1 new feature, 189
version 16.0 behavior change, 35
version 16.0 enhancement, 28
CREATE ENCRYPTED FILE statement
version 16.0 behavior change, 35
version 16.0 enhancement, 28
CREATE ENCRYPTED TABLE statement
version 16.0 behavior change, 35
CREATE EVENT statement
version 11.0.0 enhancement, 222
version 16.0 enhancement, 26
version 16.0 new feature, 17
CREATE EXISTING TABLE statement
version 12.0.1 enhancement, 86
CREATE EXTERNLOGIN statement
version 16.0 behavior change, 35
CREATE FUNCTION statement
version 11.0.1 enhancement, 190
version 12.0.0 enhancement, 132
version 16.0 enhancement, 26
create function wizard
version 12.0.0 enhancement, 178
CREATE INDEX statement
version 12.0.0 enhancement, 130, 177
version 16.0 enhancement, 26
CREATE LDAP SERVER statement
version 16.0 new feature, 12
CREATE LOGIN POLICY statement
version 11.0.0 new feature, 220
version 16.0 behavior change, 7, 12
create maintenance plan wizard
version 11.0.1 enhancement, 201
version 12.0.0 enhancement, 178
CREATE MATERIALIZED VIEW statement
version 11.0.0 enhancement, 221
CREATE MIRROR SERVER statement
version 12.0.0 new feature, 118
create passthrough download wizard
version 11.0.0 new feature, 258
create passthrough script wizard
version 11.0.0 new feature, 258
CREATE PROCEDURE statement
version 11.0.1 enhancement, 190
version 12.0.0 behavior change, 143
version 12.0.0 enhancement, 132
version 12.0.1 enhancement, 86
version 16.0 enhancement, 26
Create Project Wizard
version 16.0 new feature, MobiLink plug-in, 48
CREATE PUBLICATION statement
version 12.0.0 enhancement, 131
CREATE ROLE statement
version 16.0 new feature, 9
create sequence generator wizard
version 12.0.0 new feature, 119
CREATE SEQUENCE statement
version 12.0.0 new feature, 119
CREATE SERVER statement
version 12.0.0 enhancement, 130
version 12.0.1 enhancement, 86
version 16.0 enhancement, 26
create service wizard
version 11.0.1 enhancement, 201
CREATE SPATIAL REFERENCE SYSTEM statement
version 12.0.0 new feature, 115
create spatial reference system wizard
version 12.0.0 new feature, 116
CREATE SPATIAL UNIT OF MEASURE statement
version 12.0.0 new feature, 115
CREATE statement
version 16.0 enhancement, 9
Create Synchronization Model Wizard
version 16.0 enhancement, MobiLink plug-in, 49
CREATE SYNCHRONIZATION PROFILE statement
version 12.0.0 new feature, 130
create synchronization profile wizard
version 11.0.0 new feature, 257
CREATE SYNCHRONIZATION SUBSCRIPTION statement
version 12.0.0 enhancement, 150
version 12.0.0 enhancement, MobiLink, 154
CREATE TABLE statement
version 11.0.1 enhancement, 190
version 12.0.0 enhancement, 131
CREATE TEMPORARY TRACE EVENT SESSION statement
version 16.0 new feature, 26
CREATE TEMPORARY TRACE EVENT statement
version 16.0 new feature, 27
create text configuration object wizard
version 12.0.0 enhancement, 177
CREATE TEXT CONFIGURATION statement
version 11.0.0 new feature, 220
CREATE TEXT INDEX statement
version 11.0.0 new feature, 221
version 12.0.0 enhancement, 131
version 16.0 enhancement, 26
CREATE TRIGGER statement
troubleshooting version 16 upgrades, 282
version 11.0.1 enhancement, 190
version 12.0.0 enhancement, 132
create unit of measure wizard
version 12.0.0 new feature, 116
CREATE USER statement
version 11.0.0 new feature, 220
create user wizard
version 11.0.0 enhancement, 260
CREATE VARIABLE statement
version 12.0.0 enhancement, 130, 132
CREATE VIEW statement
version 11.0.1 enhancement, 190
version 12.0.0 enhancement, 132
version 16.0 behavior change, 33, 39
createcert utility
version 16.0 behavior change, 3
version 16.0 enhancement, 22
createkey utility
version 11.0.0 new feature, 247
CROSS APPLY clause
version 11.0.0 new feature, 222
CURRENT UTC TIMESTAMP special value
version 12.0.0 behavior change, 141
current_download_row_count
version 16.0 enhancement, UltraLite C/C++, 61
version 16.0 enhancement, UltraLite.NET, 61
CurrentMultiProgrammingLevel property
version 12.0.0 new feature, 127
Cursor property
version 12.0.1 new feature, 91
CursorOpen property
version 12.0.1 new feature, 91
cursors
  version 12.0.0 enhancement, 127
CustDB
  version 11.0.0 behavior change, UltraLite, 255
CustDB sample
  version 16.0 enhancement, UltraLiteJ, 64
  version 16.0 removed feature, UltraLite, 65
CustDB UltraLite sample
  version 11.0.0 behavior change, 255
custom authentication messages
  version 16.0 enhancement, UltraLite, 60
custom collations
  only preserved with internal unload, 281
customized secure feature keys
  version 16.0 behavior change, 34

d
daemon
  version 12.0.0 behavior change, 114
data scripts
  version 12.0.0 behavior change, MobiLink, 158
  version 16.0 new feature, MobiLink, 42
data sources
  version 12.0.0 enhancement, 122
data type conversions
  version 12.0.0 enhancement, 138
data types
  version 12.0.0 new feature, 133
  version 16.0 enhancements, 28
database cleaner
  version 12.0.0 behavior change, 144
  version 12.0.1 behavior change, 89
database documentation wizard
  version 11.0.0 new feature, 257
  version 12.0.0 enhancement, 177
database encryption
  version 11.0.0 enhancement, 210
database file format
  upgrading, 280, 286
database mirroring
  applying Support Packages, 293
  upgrading, 290
  upgrading (database servers) to minor releases, 291
  upgrading databases, 295
  version 11.0.0 behavior change, 236
  version 11.0.0 enhancement, 209
version 11.0.1 behavior change, 196
version 12.0.0 behavior change, 142
version 12.0.0 enhancement, 118
version 12.0.1 enhancement, 75
database options
  version 11.0.0 enhancement, 206
  version 12.0.1 behavior change, OnDemand, 74
  version 16.0 enhancements, 22
database properties
  version 11.0.0 new feature, 216
  version 12.0.0 new feature, 90, 126
  version 12.0.1 enhancement, 76
  version 16.0 enhancements, 24
database server messages window
  version 12.0.0 enhancement, 185
database server properties
  version 12.0.0 new feature, 127
  version 12.0.1 enhancement, 76
  version 12.0.1 new feature, 91
  version 16.0 enhancements, 24
database sizes
  version 11.0.0 enhancement, 229
database tools import libraries
  version 11.0.0 new feature, 235
database validation on Android
  version 16.0 enhancement, UltraLiteJ, 63
database worker threads
  version 12.0.1 new feature, MobiLink, 94
DatabaseInfo.getNumberRowsToUpload method [UltraLiteJ]
  version 12.0.1 enhancement, 80
DatabaseInfo.getPageSize method [UltraLiteJ]
  version 12.0.1 enhancement, 80
DatabaseInfo.getRelease method [UltraLiteJ]
  version 12.0.1 enhancement, 80
DatabaseManager.createFileTransfer method [UltraLiteJ]
  version 12.0.1 enhancement, 77
DatabaseMetaData
  version 16.0 behavior change, 38
DatabaseName connection parameter
  version 12.0.1 behavior change, OnDemand, 74
databases
  comparing, version 12.0.1 new feature, 75
  rebuilding in Sybase Central, 277
  rebuilding using dbunload utility, 278
  rebuilding version 9 and earlier, 283
rebuilding version 9 and earlier using dbunload, 285
upgrading, 272
upgrading file format from version 10.0.0 and later, 286
upgrading file format from version 9 and earlier, 280
upgrading in a database mirroring system, 295
upgrading version 10 and later using ALTER DATABASE, 289
upgrading version 10 and later using dbupgrad, 288
upgrading version 10 and later using Sybase Central, 286
DATEADD function
version 12.0.0 enhancement, 129
DATEDIFF function
version 12.0.0 behavior change, 144
version 12.0.0 enhancement, 129
DATENAME function
version 12.0.0 enhancement, 129
DATEPART function
version 12.0.0 enhancement, 129
dates
version 12.0.0 enhancement, 129
DATETIMEOFFSET data type
version 12.0.0 new feature, 133
DateTimeWithTimeZone class
version 12.0.0 new feature, MobiLink, 152
DB_EXTENDED_PROPERTY function
version 12.0.0 enhancement, 129
version 12.0.1 enhancement, 85
version 16.0 behavior change, 36
DB_NAME function
version 16.0 behavior change, 36
DB_PROPERTY function
version 16.0 behavior change, 36
db_publisher property
version 16.0 new feature, 8
DBA
version 12.0.1 enhancement, 75
DBA authority
version 16.0 deprecated, 40
dBase formats, INPUT statement
version 11.0.0 removed, support, 262
dBase formats, OUTPUT statement
version 11.0.0 removed, support, 262
dbackup utility
version 16.0 enhancement, 19
dbcapi.dll
version 11.0.0 new feature, 223
dbconsole utility
version 11.0.0 enhancement, 263
dbeng12
version 12.0.0 behavior change, 114, 141
version 12.0.1 behavior change, 92
dbfhide utility
version 12.0.0 enhancement, 124
dbinfo utility
version 12.0.0 enhancement, 139
dbinit utility
version 12.0.0 behavior change, 145
version 16.0 enhancement, 31, 32
dbqsl utility
version 11.0.0 behavior change, 240
version 16.0 new feature, 69
dbsql.exe
version 11.0.0 behavior change, 263
dbsqlc utility
version 11.0.0 behavior change, 262
version 11.0.1 behavior change, 196
dbsqlg
version 11.0.0 renamed, 263
dblgen12.dll registry entry
version 12.0.0 enhancement, 140
DBLIB
version 11.0.0 new feature, 235
dblibtb.dll
version 11.0.0 new feature, 235
dblibtw.dll
version 11.0.0 new feature, 235
dblic utility
version 12.0.0 enhancement, 124
version 12.0.1 enhancement, 93
dblog utility
version 12.0.0 behavior change, 114
dbltm utility
version 12.0.0 behavior change, 114
version 12.0.0 unsupported, 113
dbmlsync -sp option
version 11.0.0 new feature, 247
dbmlsync APIs for .NET and C++
version 11.0.0 new feature, 247
dbmlsync integration component
version 11.0.0 deprecated, 248
version 12.0.0 removed, 161
dbmlsync message log
version 12.0.1 enhancement, MobiLink, 83
dbmlsync message log scanning
  version 11.0.0 enhancement, 247
dbmlsync StreamCompression extended option
  version 11.0.0 unsupported, 248
dbmlsync
  version 12.0.0 behavior change, 114
dbmodenv
  version 11.0.1 behavior change, 196
dbn12 utility
  version 12.0.0 behavior change, 114
dbremote utility
  version 12.0.0 behavior change, 114
  version 16.0 behavior change, SQL Remote, 72
DBSC_CancelRet enumeration
  version 12.0.0 new feature, MobiLink, 155
DBSC_ERR_ACTIVE_SYNC_NOT_CANCELED member
  version 12.0.0 new feature, MobiLink, 155
DBSC_ERR_DEAD_SERVER member
  version 12.0.0 new feature, MobiLink, 155
DBSC_Event structure
  version 16.0 new feature, Dbmlsync .NET API, 47
  version 16.0 new feature, Dbmlsync C++ API, 47
DBSC_EVENTTYPE_DOWNLOAD_COMMITTED
  version 16.0 new feature, Dbmlsync .NET API, 47
  version 16.0 new feature, Dbmlsync C++ API, 47
DBSC_EVENTTYPE_DOWNLOAD_START event
  version 16.0 new feature, Dbmlsync .NET API, 48
  version 16.0 new feature, Dbmlsync C++ API, 47
DBSC_EVENTTYPE_UPLOAD_SENT event
  version 16.0 new feature, Dbmlsync .NET API, 47
  version 16.0 new feature, Dbmlsync C++ API, 47
DBSC_EVENTTYPE_UPLOAD_START event
  version 16.0 new feature, Dbmlsync .NET API, 47
  version 16.0 new feature, Dbmlsync C++ API, 47
dbspaces
  version 11.0.0 behavior change, 236, 240
dbspawn utility
  version 16.0 enhancement, 22
dbsrv12
  version 12.0.0 behavior change, 114
dbstats utility
  version 12.0.0 new feature, 137
dbsupport utility
  version 11.0.0 enhancement, 265
  version 12.0.0 behavior change, 114
version 12.0.0 enhancement, 124
version 16.0 behavior change, 35
dbsvc utility
  version 11.0.1 enhancement, 188
  version 12.0.0 behavior change, 113, 147
  version 12.0.0 enhancement, 124
  version 12.0.0 new feature, 138
  version 16.0 behavior change, 15
dbtlstb.dll
  version 11.0.0 new feature, 235
dbtlstw.dll
  version 11.0.0 new feature, 235
dbtran utility
  version 11.0.0 enhancement, 212
  version 12.0.0 behavior change, 113
DBTYPE_DBTIME
  Borland Delphi, 41
DBTYPE_DBTIME2
  version 16.0 enhancement, 28
DBTYPE_DBTIMESTAMPOFFSET
  Borland Delphi, 41
  version 12.0.1 enhancement, 87
dbunload utility
  rebuilding version 10 and later databases, 278
  rebuilding version 9 and earlier databases, 285
  troubleshooting version 16 upgrades, 282
  version 11.0.0 behavior change, 239
  version 11.0.0 enhancement, 212
  version 11.0.1 enhancement, 188
  version 16.0 enhancement, 22
dbupgrad utility
  upgrading version 10 and later databases, 288
  version 11.0.1 behavior change, 196
  version 16.0 enhancement, 18, 33
dbvalid utility
  version 11.0.0 enhancement, 212
dbxtract utility
  version 11.0.0 behavior change, 239
  version 11.0.0 enhancement, 249
  version 12.0.0 enhancement, 124, 162
DCX
  version 11.0.1 new feature, 202
  version 12.0.0 enhancement, 185
  version 12.0.1 enhancement, 112
Deadlock system event
  version 11.0.0 new feature, 231
dbloc utility
  version 11.0.0 enhancement, 260
DECLARE CURSOR statement
version 12.0.0 behavior change, 143

DECLARE LOCAL TEMPORARY TABLE statement
troubleshooting version 16 upgrades, 282

DECLARE statement
version 12.0.0 enhancement, 130

DECRYPT function
version 11.0.0 enhancement, 211
version 16.0 behavior change, 35
version 16.0 enhancement, 19
default database servers
version 11.0.1 enhancement, 192

DEFAULT VALUES clause
version 11.0.1 new feature, 189
defaultShowMultipleResultSets
version 16.0 new feature, 69
defaultShowResultsForAllStatements
version 16.0 new feature, 69

DELETE statement
version 11.0.0 behavior change, 238
version 11.0.0 enhancement, 223
version 11.0.1 enhancement, 191
version 12.0.0 behavior change, 141
version 12.0.0 enhancement, 133
version 12.0.1 enhancement, 87
delete_old_logs option
version 12.0.0 behavior change, 114
delimiting SQL strings
version 12.0.0 enhancement, 139

Delphi (Borland)
version 16.0 unsupported, 41
demo.db
version 11.0.0 enhancement, 265

Deploy Synchronization Model Wizard
version 16.0 behavior change, 49
deploy synchronization model wizard
version 11.0.0 enhancement, 260

deployment
version 16.0 behavior change for FIPS, 3

Deployment Wizard
version 12.0.0 enhancement, 113
deployment wizard
version 11.0.0 enhancement, 260
deprecated features
version 11.0.0, 205
version 11.0.1, 187
version 12.0.0, 113

version 12.0.1, 73
version 16.0, 1
version 9.0.2 and earlier, vii

DER encoded keys
version 12.0.1 new feature, MobiLink, 95

DESCRIBE statement
version 11.0.0 enhancement, 262
version 16.0 behavior change, 34
devices
version 11.0.0 enhancement, UltraLite, 252
version 12.0.0 enhancement, UltraLite, 163
version 12.0.1 enhancement, UltraLite, 103
version 16.0 enhancement, UltraLite, 60
diagnostic tracing
version 16.0 behavior change, 34
direct page scans
version 11.0.0 new feature, UltraLite Table API , 254
disable table editing
version 11.0.0 new feature, 258

disabling result set editing
version 11.0.0 new feature, 259

Disk Reads Work Table statistic
version 11.0.0 new feature, 216
disk sandboxing
version 16.0 new feature, 20
disk_sandbox option
version 16.0 new feature, 21
disk_sandbox property
version 16.0 new feature, 22

DiskRead property
version 11.0.0 new feature, 239
DiskReadEng property
version 11.0.0 renamed, 239
DiskReadHint property
version 11.0.0 new feature, 239
DiskReadHintPages property
version 11.0.0 new feature, 239
DiskReadHintScatterLimit property
version 11.0.0 new feature, 239
DiskReadWorkTable property
version 11.0.0 new feature, 214
DiskRetryRead property
version 11.0.0 new feature, 215
DiskRetryReadScatter
version 11.0.0 new feature, 215
DiskRetryReadScatter property
version 11.0.0 new feature, 216
<table>
<thead>
<tr>
<th>Property/Option</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiskRetryWrite property</td>
<td>version 11.0.0 new feature</td>
<td>215</td>
</tr>
<tr>
<td>DiskSandbox property</td>
<td>version 16.0 new feature</td>
<td>22</td>
</tr>
<tr>
<td>DiskSyncRead property</td>
<td>version 11.0.0 new feature</td>
<td>214</td>
</tr>
<tr>
<td>DiskSyncWrite property</td>
<td>version 11.0.0 new feature</td>
<td>214</td>
</tr>
<tr>
<td>DiskWaitRead property</td>
<td>version 11.0.0 new feature</td>
<td>214</td>
</tr>
<tr>
<td>DiskWaitWrite property</td>
<td>version 11.0.0 new feature</td>
<td>214</td>
</tr>
<tr>
<td>DiskWriteHint property</td>
<td>version 11.0.0 new feature</td>
<td>214</td>
</tr>
<tr>
<td>DiskWriteHintPages property</td>
<td>version 11.0.0 new feature</td>
<td>214</td>
</tr>
<tr>
<td>divide_by_zero_error option</td>
<td>version 11.0.0 unsupported</td>
<td>241</td>
</tr>
<tr>
<td>version 12.0.0 supported</td>
<td>143</td>
<td></td>
</tr>
<tr>
<td>DLL protocol option</td>
<td>version 11.0.0 unsupported</td>
<td>243</td>
</tr>
<tr>
<td>version 12.0.0 supported</td>
<td>143</td>
<td></td>
</tr>
<tr>
<td>DML</td>
<td>version 12.0.0 new feature, selecting from</td>
<td>121</td>
</tr>
<tr>
<td>DocCommentXchange (DCX)</td>
<td>version 11.0.1 new feature</td>
<td>202</td>
</tr>
<tr>
<td>documentation enhancements</td>
<td>version 11.0.0, 264</td>
<td></td>
</tr>
<tr>
<td>version 11.0.1, 203</td>
<td></td>
<td></td>
</tr>
<tr>
<td>version 12.0.0, 185</td>
<td></td>
<td></td>
</tr>
<tr>
<td>version 12.0.1, 112</td>
<td></td>
<td></td>
</tr>
<tr>
<td>documenting a database</td>
<td>version 11.0.0 new feature</td>
<td>257</td>
</tr>
<tr>
<td>version 12.0.0 enhancement</td>
<td>177</td>
<td></td>
</tr>
<tr>
<td>download acknowledgement</td>
<td>version 11.0.0 behavior change</td>
<td>248</td>
</tr>
<tr>
<td>download delete subset</td>
<td>version 12.0.0 new feature, MobiLink</td>
<td>148</td>
</tr>
<tr>
<td>download only tables</td>
<td>version 12.0.1 new feature, UltraLite</td>
<td>102</td>
</tr>
<tr>
<td>DriveBus property</td>
<td>version 12.0.0 new feature</td>
<td>126</td>
</tr>
<tr>
<td>DriveModel property</td>
<td>version 12.0.0 new feature</td>
<td>126</td>
</tr>
<tr>
<td>Driver connection parameter</td>
<td>version 11.0.0 enhancement</td>
<td>225</td>
</tr>
<tr>
<td>DROP CERTIFICATE statement</td>
<td>version 16.0 new feature</td>
<td>26</td>
</tr>
<tr>
<td>DROP DATABASE statement</td>
<td>version 16.0 new feature</td>
<td>27</td>
</tr>
<tr>
<td>DROP EVENT statement</td>
<td>version 11.0.0 enhancement</td>
<td>222</td>
</tr>
<tr>
<td>version 11.0.1 enhancement</td>
<td>190</td>
<td></td>
</tr>
<tr>
<td>version 12.0.0 enhancement</td>
<td>131</td>
<td></td>
</tr>
<tr>
<td>DROP FUNCTION statement</td>
<td>version 11.0.1 enhancement</td>
<td>190</td>
</tr>
<tr>
<td>version 12.0.0 enhancement</td>
<td>131</td>
<td></td>
</tr>
<tr>
<td>DROP INDEX statement</td>
<td>version 12.0.0 enhancement</td>
<td>131</td>
</tr>
<tr>
<td>DROP LDAP SERVER statement</td>
<td>version 16.0 new feature</td>
<td>12</td>
</tr>
<tr>
<td>DROP LOGIN POLICY statement</td>
<td>version 11.0.0 new feature</td>
<td>220</td>
</tr>
<tr>
<td>DROP MATERIALIZED VIEW statement</td>
<td>version 11.0.1 enhancement</td>
<td>190</td>
</tr>
<tr>
<td>version 12.0.0 enhancement</td>
<td>131</td>
<td></td>
</tr>
<tr>
<td>DROP MIRROR SERVER statement</td>
<td>version 12.0.0 new feature</td>
<td>118</td>
</tr>
<tr>
<td>DROP PROCEDURE statement</td>
<td>version 11.0.1 enhancement</td>
<td>190</td>
</tr>
<tr>
<td>version 12.0.0 enhancement</td>
<td>131</td>
<td></td>
</tr>
<tr>
<td>DROP PUBLICATION statement</td>
<td>version 12.0.0 enhancement</td>
<td>131</td>
</tr>
<tr>
<td>DROP REMOTE CONNECTION statement</td>
<td>version 16.0 new feature</td>
<td>26</td>
</tr>
<tr>
<td>DROP ROLE statement</td>
<td>version 16.0 new feature</td>
<td>10</td>
</tr>
<tr>
<td>DROP SEQUENCE statement</td>
<td>version 12.0.0 new feature</td>
<td>119</td>
</tr>
<tr>
<td>DROP SPATIAL REFERENCE SYSTEM statement</td>
<td>version 12.0.0 new feature</td>
<td>115</td>
</tr>
<tr>
<td>DROP SPATIAL UNIT OF MEASURE statement</td>
<td>version 12.0.0 new feature</td>
<td>115</td>
</tr>
<tr>
<td>drop synchronization model</td>
<td>version 16.0 new feature, MobiLink plug-in</td>
<td>49</td>
</tr>
<tr>
<td>DROP TABLE statement</td>
<td>version 11.0.1 enhancement</td>
<td>190</td>
</tr>
<tr>
<td>version 12.0.0 enhancement</td>
<td>131</td>
<td></td>
</tr>
<tr>
<td>DROP TEXT CONFIGURATION statement</td>
<td>version 11.0.0 new feature</td>
<td>221</td>
</tr>
<tr>
<td>DROP TEXT INDEX statement</td>
<td>version 11.0.0 new feature</td>
<td>221</td>
</tr>
<tr>
<td>DROP TRACE EVENT SESSION statement</td>
<td>version 16.0 new feature</td>
<td>27</td>
</tr>
<tr>
<td>DROP TRACE EVENT statement</td>
<td>version 16.0 new feature</td>
<td>27</td>
</tr>
</tbody>
</table>
Index

DROP TRIGGER statement
version 11.0.1 enhancement, 190
version 12.0.0 enhancement, 131

DROP USER statement
version 11.0.0 new feature, 220

DROP VARIABLE statement
version 11.0.1 enhancement, 190
version 12.0.0 enhancement, 131

DROP VIEW statement
version 11.0.1 enhancement, 190
version 12.0.0 enhancement, 131

DropBadStatistics property
version 12.0.0 new feature, 128

DropUnusedStatistics property
version 12.0.0 new feature, 128
dual control passwords
version 16.0 new feature, 18
duplicate text indexes
version 11.0.1 behavior change, 194
dynamic cache sizing
version 12.0.1 behavior change, 92

EnableFlush registry entry
version 12.0.0 new feature, 140

ENAME protocol option
version 11.0.0 unsupported, 241

ENCRYPT function
version 11.0.0 enhancement, 211
version 16.0 behavior change, 35
version 16.0 enhancement, 19
encrypt_aes_random_iv option
version 11.0.0 unsupported, 243

ENCRIPT_PASSWORD connection property
version 11.0.0 new feature, 211

encryption
version 11.0.0 enhancement, 210, 228
version 12.0.0 enhancement, 129
version 16.0 enhancement, 30
version 16.0 new feature, UltraLite, 59

ENCRYPTION connection parameter
version 16.0 deprecated, 13

end-to-end encryption
version 11.0.0 enhancement, UltraLite, 252
version 11.0.0 new feature, 247

end_synchronization scripts
version 16.0 behavior change, 50
endianness
version 11.0.0 enhancement, 228
version 11.0.1 behavior change, 196

ENG connection parameter
version 12.0.0 deprecated, 147

EngineName connection parameter
version 12.0.0 deprecated, 147

Entity Framework 5.0 support
version 16.0 new feature, 28

Entity Framework support
version 11.0.1 new feature, 191
version 12.0.1 new feature, 77

environment.plist
version 11.0.1 behavior change, 196

ER diagram
version 16.0 enhancement, 68

error codes
version 12.0.0 new feature, Relay Server, 156
error messages
version 12.0.0 new feature, Relay Server, 156
version 12.0.1 new feature, UltraLite, 107
error reporting
  version 11.0.0 enhancement, 265
  version 12.0.0 enhancement, 124
derror_handler
  version 16.0 behavior change, 51
Esc key
  version 12.0.0 behavior change, 182
escape characters
  version 12.0.0 enhancement, utilities, 113
  version 12.0.1 behavior change, 92
Escape connection parameter
  version 12.0.0 new feature, 122
ESTIMATE_SOURCE function
  version 12.0.0 enhancement, 139
ETD files
  version 16.0 new feature, 16
event trace data (ETD) file management utility (dbmanageetd)
  version 16.0 new feature, 16
event trace data files
  version 16.0 new feature, 16
event tracing
  version 16.0 new feature, 16
  version 16.0 new feature, MobiLink, 43
EVENT_PARAMETER function
  version 11.0.0 enhancement, 218
events
  version 11.0.1 enhancement, 201
EventTypeDesc
  version 11.0.0 new feature, 215
EvenTypeName
  version 11.0.0 new feature, 215
EXCEL format, INPUT statement
  version 11.0.0 removed, support, 262
EXCEL format, OUTPUT statement
  version 11.0.0 removed, support, 262
EXCEPT clause
  version 11.0.0 behavior change, 238
  version 11.0.0 enhancement, 223
execute SQL statements
  version 11.0.0 enhancement, Interactive SQL, 261
executeBatch
  version 12.0.0 enhancement, 135
execution plans
  version 11.0.0 enhancement, 231
  version 11.0.1 enhancement, 191
export wizard
  version 11.0.0 enhancement, 263
expressions
  version 12.0.0 enhancement, 138
ExtendedName protocol option
  version 11.0.0 unsupported, 241
extern_login_credentials option
  version 16.0 new feature, 23
extern_login_credentials property
  version 16.0 new feature, 24
external logins
  version 12.0.0 behavior change, 144
external unloads
  version 11.0.0 enhancement, 230
extract database wizard
  version 11.0.0 enhancement, UltraLite, 251
extraction utility (dbxtract)
  version 11.0.0 behavior change, 239
  version 11.0.0 enhancement, 249
  version 12.0.0 enhancement, 162

F
failed rebuilds
  upgrading databases to version 16, 282
failover
  version 11.0.0 new feature, MobiLink server farm, 245
FAQ
  version 12.0.0 new feature, 185
fast launchers
  version 11.0.0 enhancement, 256
  version 12.0.0 enhancement, 176
  version 16.0 enhancement, 69
favorites list
  version 11.0.0 new feature, 258
  version 12.0.0 enhancement, 180
fibers
  version 12.0.0 behavior change, 141
file formats
  upgrading, 280
file hiding utility (dbfhide)
  version 12.0.0 enhancement, 124
files
  version 11.0.0 new feature, querying, 209
FileTransfer interface [UltraLiteJ]
  version 12.0.1 enhancement, 77
FileTransferProgressData interface [UltraLiteJ]
  version 12.0.1 enhancement, 77
version 12.0.0 behavior change, MobiLink, 157

H

HANA remote data access
version 16.0 new feature, 17
HANAODBC server class
version 16.0 new feature, 17
HasEndianSwapFix property
version 11.0.0 new feature, 216
HASH function
version 12.0.0 enhancement, 128
hash sizes
version 16.0 new feature, UltraLite, 59
HasTornWriteFix property
version 12.0.0 new feature, 126
header_limit option
version 16.0 new feature, MobiLink, 44
health and statistics
version 11.0.0 new feature, 257
HeapsCarver property
version 12.0.1 new feature, 90
HeapsLocked property
version 12.0.1 new feature, 90
HeapsQuery property
version 12.0.1 new feature, 90
HeapsRelocatable property
version 12.0.1 new feature, 90
HEXTOBIN function
version 16.0 new feature, 25
hooks
version 16.0 behavior change, MobiLink, 52
Host connection parameter
version 12.0.0 new feature, 119
host protocol option
version 12.0.0 behavior change, 142
hosted Relay Server
version 11.0.0 new feature, 245
HTTP
version 11.0.0 behavior change, 236
HTTP authentication
version 16.0 new feature, UltraLite, 58
HTTP messaging system
version 12.0.1 behavior change, 77
HTTP performance improvements
version 16.0 new feature, MobiLink, 47
HTTP server
version 16.0 behavior change, 34
HTTP service
version 12.0.0 enhancement, 134
http_buffer_responses protocol option
version 12.0.0 new feature, MobiLink, 154
http_connection_pool_basesize option
version 12.0.0 new feature, 125
http_connection_pool_timeout option
version 12.0.0 new feature, 125
http_password option
version 12.0.1 new feature, Relay Server, 98
http_proxy_password option
version 12.0.1 new feature, Relay Server, 98
http_proxy_userid option
version 12.0.1 new feature, Relay Server, 98
HTTP_RESPONSE_HEADER function
version 12.0.0 new feature, 129
http_userid option
version 12.0.1 new feature, Relay Server, 98
HTTP_VARIABLE function
version 12.0.0 new feature, 129
HttpAddresses property
version 11.0.0 new feature, 215
HttpConnPoolCachedCount property
version 12.0.0 new feature, 126
HttpConnPoolHits property
version 12.0.0 new feature, 126
HttpConnPoolMisses property
version 12.0.0 new feature, 126
HttpConnPoolSteals property
version 12.0.0 new feature, 126
HttpNumActiveReq property
version 11.0.0 new feature, 215
HttpNumConnections property
version 11.0.0 new feature, 215
HttpNumSessions property
version 11.0.0 new feature, 215
HTTPS
version 11.0.0 enhancement, 228
HttpsAddresses property
version 11.0.0 new feature, 215
HttpsNumActiveReq property
version 11.0.0 new feature, 215
HttpsNumConnections property
version 11.0.0 new feature, 215
hyphens
version 11.0.1 behavior change, full text search, 193
Index

iAnywhere JDBC driver
  version 12.0.0 deprecated, 147
iAnywhere Solutions 12 - Oracle ODBC driver
  version 12.0.0 enhancement, 150
iAnywhere Solutions Oracle driver
  version 11.0.0 enhancement, 244
  version 16.0 behavior change, 13
iastor registry entry
  version 12.0.0 new feature, 140
iastorv registry entry
  version 12.0.0 new feature, 140
IBM DB2 LUW
  version 16.0 new feature, MobiLink, 44
IBM DB2 LUW 8.2
  version 12.0.0 discontinued, MobiLink, 160
IBM DB2 LUW 9.5 support
  version 16.0 removed, MobiLink, 53
IBM DB2 Mainframe
  version 12.0.0 discontinued, MobiLink, 160
  identifiers
    version 12.0.0 enhancement, 130, 139, 143
    version 16.0 enhancement, 31
IdentifySimulatedClient callback
  version 12.0.1 behavior change, MobiLink (mlgenreplayapi), 99
identity files
  version 11.0.0 behavior change, 266
identity forwarding
  version 12.0.1, Relay Server, 83
identity synchronization parameter
  version 12.0.0 new feature, MobiLink, 154
identity_password synchronization parameter
  version 12.0.0 new feature, MobiLink, 154
Idle connection parameter
  version 12.0.0 behavior change, 142
IF EXISTS clause
  version 11.0.0 enhancement, 190
  version 12.0.0 new feature, 131
  version 12.0.0 new feature, UltraLiteJ, 164
IF expressions
  version 11.0.0 enhancement, 222
IF NOT EXISTS clause
  version 11.0.1 new feature, 190
  version 12.0.0 new feature, 131
  version 12.0.0 new feature, UltraLiteJ, 164
IF statement
  version 11.0.0 enhancement, 222
  version 11.0.0 enhancement, UltraLite, 253
  ignored deletes
    version 16.0 new feature, UltraLite, 57
  ignored updates
    version 16.0 new feature, UltraLite, 57
  ignoring scripts
    version 11.0.1 new feature, MobiLink (mlsrv11), 197
IIS
  version 16.0, 32-bit Relay Server for Windows discontinued, 57
IIS 6
  version 12.0.1 quick deployment, Relay Server, 83
IIS 7
  version 12.0.1 quick deployment, Relay Server, 83
IIS 8
  version 16.0 new feature, supported for Relay Server, 54
IMMEDIATE clause
  version 11.0.1 enhancement, MESSAGE statement, 190
  immediate views
    version 11.0.0 new feature, 208
    version 16.0 enhancement, 26
Import Wizard
  version 16.0 enhancement, 70
import wizard
  version 11.0.0 enhancement, 263
  version 12.0.1 enhancement, 84
in memory mode
  version 11.0.0 new feature, 213
index hints
  version 12.0.0 behavior change, 145
  version 12.0.0 enhancement, 130
INDEX ONLY clause
  version 11.0.0 new feature, 222
indexes
  version 11.0.0 behavior change, 240
  version 11.0.0 enhancement, 206
  version 11.0.0 enhancement, performance, 229
  version 12.0.0 enhancement, 138
  version 16.0 behavior change, 33, 34
  version 16.0 enhancement, 22
indexing performance improvements
  version 16.0 new feature, UltraLite, 59
indicator variables
  version 12.0.0 unsupported, short int, 147
information utility (dbinfo)
  version 12.0.0 enhancement, 139
initial cache size
  version 12.0.1 behavior change, 92
initialization utility (dbinit)
  version 12.0.0 behavior change, 145
  version 12.0.0 enhancement, 124, 140
  version 16.0 enhancement, 31, 32
InList algorithm (IN)
  version 11.0.0 enhancement, 229
INPUT INTO statement
  version 11.0.0 behavior change, 262
INPUT statement
  version 11.0.0 behavior change, 262
  version 11.0.0 enhancement, 262
  version 12.0.0 enhancement, 182
  version 12.0.1 enhancement, 84
INSERT statement
  multi-row enhancement, 132
  version 11.0.0 behavior change, 238
  version 11.0.0 enhancement, 223, 229
  version 11.0.1 enhancement, 189
  version 12.0.0 enhancement, 132
installation directory
  version 11.0.0 behavior change, 266
INTEGER UNSIGNED data type
  version 16.0 new feature, 28
integrated outbound enabler
  version 12.0.0 new feature, MobiLink, 149
Intel storage driver
  version 12.0.0 enhancement, 140
Interactive SQL
  auto-complete, 108
  text completion, 108
  version 11.0.0 enhancement, 260
  version 11.0.0 new feature, 258
  version 12.0.0 enhancement, 182
  version 12.0.0 new feature, 180
  version 12.0.1 enhancement, 84, 109
  version 12.0.1 new feature, 108
  version 16.0 behavior change, 70
  version 16.0 new feature, 69
Interactive SQL behavior changes
  version 11.0.0, 259
  version 11.0.1, 202
  version 12.0.0, 181
  version 12.0.1.1, 109
Interactive SQL error window
ISENCRYPTED function
  version 12.0.0 new feature, 129
isolation level
  version 11.0.0 enhancement, UltraLite
    ReadUncommitted, 251
isolation level 2
  version 12.0.0 enhancement, 138
isolation level 3
  version 12.0.0 enhancement, 138
IsPortableDevice property
  version 12.0.0 new feature, 127
isql_allow_client_file_read
  version 11.0.0 renamed, 263
isql_allow_client_file_write
  version 11.0.0 renamed, 263
isql Allow_read_client_file option
  version 11.0.0 renamed , 263
isql_allow_write_client_file option
  version 11.0.0 renamed, 263
isql_command_timing option
  version 12.0.0 enhancement, 180
isql_plan option
  version 11.0.0 removed, 263
ISYSCAPABILITYNAME
  version 11.0.0 behavior change, 232
ISYSCERTIFICATE
  version 16.0 new feature, 29
ISYSCOLSTAT
  version 16.0 enhancement, 29
ISYSDBFILe
  version 11.0.0 behavior change, 232
ISYSDBSPACE
  version 11.0.0 behavior change, 232
ISYSDBSPACEPERM
  version 11.0.0 behavior change, 232
ISYSEVENTTYPE
  version 11.0.0 behavior change, 232
ISYSEXTERNENVOBJECT
  version 16.0 enhancement, 29
ISYSFILE
  version 11.0.0 behavior change, 232
ISYGROLEGRANT
  version 16.0 new feature, 10
ISYGROLEGRANTEXT
  version 16.0 new feature, 10
ISYSHISTORY
  version 16.0 enhancement, 29
ISYSIDX
version 16.0 enhancement, 29
ISYSUSERTYPE
version 12.0.0 enhancement, 137
ISYSVIEW
version 11.0.0 behavior change, 232
version 16.0 enhancement, 29
Itanium
version 11.0.0 behavior change, 238

J
Java and .NET scripts returning SQL
version 12.0.0 discontinued, MobiLink, 160
Java DownloadTableData interface
version 11.0.0 enhancement, 245
Java in the database
version 12.0.1 enhancement, 88
Java ME
version 11.0.0 new feature, UltraLite support on Java ME, 254
version 16.0 removed feature, UltraLite support on Java ME, 66
Java SE
version 11.0.0 new feature, UltraLite support on Java SE, 254
java_class_path option
version 12.0.1 new feature, 84
java_location
version 16.0 deprecated, 40
java_main_userid
version 16.0 deprecated, 40
java_main_userid property
version 12.0.0 unsupported, 146
jConnect
version 11.0.0 enhancement, 211
JDBC
troubleshooting, 297
version 12.0.1 enhancement, 88
version 16.0 behavior change, 38
JDBC 3.0
version 12.0.0 enhancement, 134
version 15 deprecated, 94
JDBC 3.0 driver
version 16.0 unsupported, 41
JDBC 4.0
version 12.0.0 enhancement, 134
version 12.0.1 enhancement, 77
JDBC drivers
version 12.0,0 behavior changes, 141
version 12.0.0 enhancement, 134
JDBC-based server classes
version 12.0.0 deprecated, 146
version 16.0 unsupported, 41
JRE
version 16.0 enhancement, 6
jre170
version 16.0 enhancement, 6
JSON services
version 16.0 behavior change, 39

K
Kerberos
version 12.0.1 behavior change, 93
key pair generator utility (createkey)
version 11.0.0 new feature, 247
keyboard shortcuts
version 11.0.0 new feature, 256
version 12.0.0 behavior change, 182

L
LAST_VALUE function
version 11.0.1 enhancement, 189
LastCheckpointTime property
version 12.0.0 new feature, 126
LastCommitRedoPos property
version 12.0.1 new feature, 76
version 16.0 new feature, 24
LastStatement property
version 16.0 behavior change, 34
LastSyncedRedoPos property
version 12.0.1 new feature, 76
version 16.0 new feature, 24
LastWrittenRedoPos property
version 12.0.1 new feature, 76
version 16.0 new feature, 24
latency
Relay Server, improving, 55
LazyClose connection parameter
version 11.0.0 behavior change, 235
LCLOSE connection parameter
version 11.0.0 behavior change, 235
LDAP
version 16.0 new feature, 11
version 16.0 new feature, MobiLink, 42
version 16.0 new feature, MobiLink plug-in, 50
LDAP user authentication
  version 16.0 enhancements, 11
  version 16.0 new feature, 11
ldap_auto_failback_period option
  version 16.0 new feature, 11
ldap_failover_to_std option
  version 16.0 new feature, 11
ldap_primary_server option
  version 16.0 new feature, 11
ldap_refresh_dn option
  version 16.0 new feature, 11
ldap_secondary_server option
  version 16.0 new feature, 11
LEN function
  version 11.0.0 enhancement, 228
LENGTH function
  version 11.0.0 enhancement, 228
licenses
  version 12.0.0 enhancement, 124
licensing
  version 16.0 behavior change, 35
Light weight polling action variables
  version 11.0.1 enhancement, 198
Light weight polling listener keywords
  version 11.0.1 enhancement, 198
LIMIT clause
  version 12.0.0 new feature, 132
  version 12.0.1 enhancement, 87
limits
  version 11.0.0 increases, 231
line numbers
  version 11.0.0 enhancement, Interactive SQL, 260
Linux
  version 11.0.0 behavior change, 238
  version 11.0.0 enhancement, 228
  version 11.0.0 new feature, 228
  version 12.0.0 enhancement, 137
  version 16.0 enhancement, 30
  version 16.0, 32-bit Relay Server discontinued, 57
little endian UTF-16 encoding
  version 11.0.0 new feature, 228
liveness timeout
  version 16.0 enhancement, MobiLink (mlreplay), 45
load balancing,
  version 11.0.0 new feature, MobiLink server farm, 245
LOAD privilege
  version 16.0 new feature, 7
LOAD STATISTICS statement
  version 12.0.0 behavior change, 141
LOAD TABLE statement
  version 11.0.0 enhancement, 219, 222, 249
  version 12.0.0 enhancement, 131
  version 12.0.1 behavior change, 92
  version 12.0.1 enhancement, 74
  version 16.0 behavior change, 39
  version 16.0 enhancement, 27
loading
  version 11.0.0 new feature, from a column, 207
  version 11.0.0 new feature, from a value, 207
  version 11.0.0 new feature, from the transaction log, 208
local temporary tables
  version 16.0 enhancement, 22
localized versions of SQL Anywhere
  version 16.0 enhancement, 30
LOCK FEATURE statement
  version 12.0.0 new feature, 130
LOCK TABLE statement
  version 12.0.0 enhancement, 138
LockIndexID property
  version 11.0.0 new feature, 214
locking
  version 11.0.0 behavior change, 239
  version 12.0.0 behavior change, 143
  version 12.0.0 enhancement, 138
  version 16.0 enhancement, MobiLink, 43
LockRowID property
  version 11.0.0 new feature, 214
locks
  version 11.0.0 new feature, Interactive SQL, 258
  version 11.0.1 behavior change, Interactive SQL, 202
log transfer manager utility
  version 12.0.0 unsupported, 113
log transfer manager utility (dbltm)
  version 12.0.0 behavior change, 114
log translation utility
  version 12.0.0 behavior change, 113
log translation utility (dbtran)
  version 11.0.0 enhancement, 212
log verbosity for targeted users
  version 11.0.1 enhancement, 198
log4j.jar
  version 12.0.0 behavior change, MobiLink, 157
LogFormat protocol option
  version 12.0.1 behavior change, 76
login policies
  version 11.0.0 new feature, 206
  version 12.0.1 enhancement, 75
  version 16.0 enhancement, 11, 18, 68
login_mode option (see version 12.0.1 new feature, OnDemand)
  version 16.0 enhancement, 12
login_procedure option
  version 11.0.0 enhancement, 213
long text plans
  version 16.0 enhancement, 31
LOTUS format, INPUT statement
  version 11.0.0 removed, support, 262
LOTUS format, OUTPUT statement
  version 11.0.0 removed, support, 262
LTM (log transfer manager utility)
  version 12.0.0 unsupported, 113
LTMGeneration property
  version 12.0.0 behavior change, 113
LTMTrunc property
  version 12.0.0 behavior change, 113

M
Mac OS X
  rebuilding databases for SQL Anywhere 16, 274
  version 12.0.0 behavior change, 145
  version 12.0.0 enhancement, 137
  version 16.0 enhancement, 30
maintenance plans
  version 12.0.0 enhancement, 179
MAPI
  version 16.0 behavior change, 37
MAPI and SMTP system procedures
  version 16.0 behavior change, 37
MAPI message type
  version 11.0.0 unsupported, 249
MapPages property
  version 11.0.0 unsupported, 240
materialized views
  upgrade considerations, 273
  version 11.0.0 enhancement, 208
  version 12.0.0 enhancement, 121
  version 12.0.0, SQL Remote initializing in extracted databases, 162
  version 16.0 enhancement, 26
MatView connection parameter
  version 12.0.1 behavior change, 92
  version 12.0.1 new feature, 84
max_priority option
  version 11.0.0 new feature, 213
max_priority property
  version 11.0.0 new feature, 214
max_retries control parameter
  version 12.0.1 enhancement, 101
MaxEventType
  version 11.0.0 new feature, 215
MaxMessage property
  version 11.0.0 deprecated, 241
MaxMultiProgrammingLevel property
  version 12.0.0 new feature, 127
MaxRemoteCapability property
  version 11.0.0 new feature, 215
MaxRequestVars protocol option
  version 16.0 new feature, 18
MAXVARS protocol option
  version 16.0 new feature, 18
MEDIAN function
  version 12.0.0 new feature, 128
memory
  version 12.0.0 enhancement, MobiLink, 149
  version 12.0.0 discontinued, MobiLink, 160
MERGE statement
  version 11.0.0 new feature, 206
message control parameters
  version 12.0.1 enhancement, 101
MESSAGE LOG clause
  version 16.0 new feature, 37
Message property
  version 11.0.0 deprecated, 241
MESSAGE statement
  version 11.0.1 enhancement, 190
MessageCategoryLimit property
  version 11.0.0 new feature, 215
MessageText property
  version 11.0.0 deprecated, 241
MessageType property
  version 11.0.0 deprecated, 241
MessageTime property
  version 11.0.0 deprecated, 241
MessageWindowSize property
  version 11.0.0 deprecated, 241
Microsoft Access
  version 11.0.0 enhancement, remote data access, 230

Copyright © 2014, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0 345
Microsoft Distributed Transaction Coordinator
version 11.0.0 new feature, 244
version 12.0.0 new feature, 150
Microsoft Excel
version 11.0.0 removed, INPUT statement support, 262
version 11.0.0 removed, OUTPUT statement support, 262
Microsoft IIS
version 16.0, 32-bit Relay Server for Windows discontinued, 57
Microsoft IIS 8
version 16.0 new feature, supported for Relay Server, 54
Microsoft SQL Server 2005 support
version 16.0 removed, MobiLink, 53
Microsoft SQL Server 2008 support
version 11.0.1 enhancement, 198
Microsoft SQL Server 2012 support
version 16.0 new feature, MobiLink, 43
Microsoft Volume Shadow Copy Service (VSS)
version 11.0.0 new feature, 210
migrate C++ applications wizard
version 11.0.0 unsupported, 255
migration
Monitor, 315
min_password_length option
version 16.0 behavior change, 33
min_role_admins option
version 16.0 new feature, 8
min_role_admins property
version 16.0 new feature, 8
minimum cache size
version 12.0.1 behavior change, 92
MinMultiProgrammingLevel property
version 12.0.0 new feature, 127
minor releases
mirroring (database servers), 291
mirror databases
version 11.0.0 enhancement, 209
MIRROR server class
version 16.0 new feature, 17
MIRROR_FILE connection parameter
version 11.0.1 enhancement, UltraLite, 199
mirroring
upgrading, 290
upgrading (database servers) to major version, 290
version 11.0.0 enhancement, 208
MirrorMode property
version 11.0.0 new feature, 216
MirrorRole property
version 12.0.0 new feature, 126
MirrorServerState property
version 12.0.0 new feature, 129
version 12.0.1 enhancement, 85
MirrorState property
version 12.0.0 new feature, 129
ml_add_certificates_file system procedure
version 16.0 new feature, MobiLink, 42
ml_add_column system procedure
version 12.0.0 behavior change, MobiLink, 158
version 16.0 deprecated, MobiLink, 54
ml_add_ldap_server system procedure
version 16.0 new feature, MobiLink, 42
ml_add_missing_dnld_scripts system procedure
version 12.0.0 new feature, MobiLink, 152
ml_add_user_auth_policy system procedure
version 16.0 new feature, MobiLink, 42
ml_column system table
version 16.0 deprecated, MobiLink, 54
ML_GET_SERVER_NOTIFICATION function
version 11.0.1 enhancement, UltraLite, 199
ml_ignore
version 11.0.1 new feature, MobiLink (mlsrv11), 197
ml_model_check_all_schema system procedure
version 16.0 new feature, MobiLink, 42
ml_model_check_version_schema system procedure
version 16.0 new feature, MobiLink, 43
ml_model_drop system procedure
version 16.0 new feature, MobiLink, 42
ml_remote_id option
version 11.0.1 enhancement, UltraLite, 199
mlarbiter utility
version 12.0.0 new feature, MobiLink, 153
mlarbiter utility for server farms
version 12.0.0 new feature, MobiLink, 149
mlarbstop utility
version 16.0 new feature, 45
mlfiletransfer
version 12.0.0 enhancement, MobiLink, 156
MLFileTransfer method
version 12.0.0 behavior change, MobiLink, 158
MLFileUpload method
version 12.0.0 new feature, MobiLink, 154
mlreplay liveness timeout
version 16.0 enhancement, MobiLink (mlreplay), 45
mlreplay utility
version 12.0.0 new feature, MobiLink, 148
version 16.0 callbacks removed, 53
mlreplay utility error messages
version 12.0.1 enhancement, MobiLink, 82
mlsrv12
version 12.0.0 behavior change, 114
version 12.0.0 new feature, MobiLink, 151
mlstop -f option
version 16.0 deprecated, MobiLink, 53
MLUser synchronization profile
version 12.0.0 deprecated, MobiLink, 161
mluser utility
version 16.0 new feature, 45
MobiLink
upgrading, 300
MobiLink 12 plug-in for Sybase Central
redesigned in version 12.0.0, 148
MobiLink agent
version 12.0.1 new feature, MobiLink, 96
MobiLink behavior changes
version 11.0.0, 248
version 11.0.1, 198
version 12.0.0, 157
version 12.0.1, 98
version 16.0, 50
MobiLink client
version 12.0.1 behavior changes, 100
MobiLink clients
version 12.0.1 new features, 98
MobiLink connectively
version 16.0 behavior change, 51
MobiLink deprecated features
version 12.0.1, 100
MobiLink discontinued features
version 12.0.1, 100
MobiLink Listener C API for Palm devices
version 12.0.0 removed, 159
MobiLink Listener utility for Palm devices
version 12.0.0 removed, 159
MobiLink log files
version 11.0.0 enhancement, UltraLite, 252
MobiLink Monitor
version 11.0.0 behavior change, 264
version 16.0 removed, MobiLink, 42, 46
MobiLink new features
version 11.0.0, 243
version 11.0.1, 197
version 12.0.0, 148
version 12.0.1, 94
version 16.0, 42
MobiLink plug-in
version 12.0.1 behavior changes, 100
version 12.0.1 enhancements, MobiLink, 97
MobiLink Profiler
version 16.0 new feature, 45
version 16.0 new feature, MobiLink, 42
MobiLink relay server
version 11.0.0 new feature, 245
MobiLink server
32-bit support, version 16.0 behavior change, 51
upgrading, 308
MobiLink server 32-bit support
version 12.0.1 behavior change, MobiLink, 98
MobiLink Server Command Line Properties window
version 16.0 new feature, MobiLink plug-in, 49
MobiLink server compatibility
version 16.0 new feature, MobiLink, 42
MobiLink server farm
version 11.0.0 new feature, 245
MobiLink system database
version 11.0.0 new feature, 244
MobiLink system objects
upgrading, 300
version 11.0.0 enhancement, 244
MobiLink system procedures
version 11.0.0 enhancement, 244
Monitor
migrating resources, 315
upgrading, 315
version 11.0.1 new feature, 203
version 12.0.0 behavior change, 184
version 12.0.0 new feature, 183
version 12.0.1 behavior change, 111
version 12.0.1 new feature, 110
version 16.0 behavior change, 70
version 16.0 new feature, 70
Monitor behavior changes
version 12.0.0, 184
version 12.0.1, 111
MSACCESSODBC server class
version 11.0.0 new feature, 230
Index

multi-threaded network processing
  version 16.0 new feature, MobiLink, 44
multiple versions
  Adaptive Server Anywhere, 275
  UltraLite, 311
multiprogramming level
  version 12.0.0 behavior change, 141
  version 12.0.0 enhancement, 120
  version 12.0.1 behavior change, 92
MySQL consolidated database
  version 11.0.0 new feature, 244
MySQL server 5.5.X support
  version 16.0 new feature, MobiLink, 44
MySQL support in MobiLink Model mode
  version 11.0.1 enhancement, 198
MYSQLODBC server class
  version 11.0.0 new feature, 230

N
Name property
  version 12.0.0 behavior change, 142
named parameters
  version 16.0 enhancement, 32
  version 16.0 new feature, MobiLink, 44
named subscription
  version 12.0.0 new feature, MobiLink, 153
named system parameters
  version 12.0.1 new feature, MobiLink, 94
NCHAR data type
  version 12.0.0 enhancement, 133
network server
  version 12.0.0 behavior change, 141
  version 12.0.0 enhancement, 120
network_conn_name variable
  version 12.0.1 new feature, MobiLink, 95
NetworkData class
  version 12.0.1 new feature, MobiLink, 82
Networking tab
  version 11.0.0 new feature, 256
new features
  version 11.0.0, 205
  version 11.0.1, 187
  version 12.0.0, 113
  version 12.0.1, 73
  version 16.0, 1
  version 9.0.2 and earlier, vii
  new_remote_id system parameter
version 12.0.1 new feature, MobiLink, 94
new_username system parameter
  version 12.0.1 new feature, MobiLink, 94
newdemo
  version 12.0.0 new feature, 139
NewPassword connection parameter
  version 11.0.0 new feature, 209
NEWPWD connection parameter
  version 11.0.0 new feature, 209
NEXT_CONNECTION function
  version 16.0 behavior change, 36
NEXT_DATABASE function
  version 16.0 behavior change, 36
NEXT_HTTP_RESPONSE_HEADER function
  version 12.0.0 new feature, 129
no_reload_status
  version 12.0.0 new feature, 134
NODE connection parameter
  version 12.0.0 new feature, 118
NodeType connection parameter
  version 12.0.0 new feature, 118
  version 12.0.1 behavior change, OnDemand, 74
  version 12.0.1 enhancement, 73
non-blocking download acknowledgement scripts
  version 11.0.0 enhancement, 246
  version 12.0.0 enhancement, 152
non-synchronized column updates
  version 12.0.1 behavior change, MobiLink, 82
non_keywords option
  version 11.0.0 behavior change, 237
notification support
  version 16.0 removed, MobiLink plug-in, 52
NOTIFY TRACE EVENT statement
  version 16.0 new feature, 27
NumLocalTempTables property
  version 16.0 new feature, 24
NVARCHAR data type
  version 12.0.0 enhancement, 133

O
ObjectType property
  version 12.0.0 new feature, 127
OData
  version 16.0 enhancement, 2
OData Producer
  version 16.0 new feature, 15
OData Server
<table>
<thead>
<tr>
<th>Feature</th>
<th>Version 16.0 Support</th>
<th>Version 12.0 Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODBC</td>
<td>behavior change, 38</td>
<td>enhancement, 2</td>
</tr>
<tr>
<td>ODBC connection parameters</td>
<td>version 12.0.0</td>
<td>enhancement, 122</td>
</tr>
<tr>
<td>ODBC data source Administrator</td>
<td>version 12.0.1</td>
<td>enhancement, OnDemand, 73</td>
</tr>
<tr>
<td>ODBC data sources</td>
<td>version 12.0.0</td>
<td>enhancement, 122</td>
</tr>
<tr>
<td>ODBC driver</td>
<td>version 12.0.0</td>
<td>enhancement, 134</td>
</tr>
<tr>
<td>ODBC driver managers</td>
<td>version 11.0.0</td>
<td>enhancement, 225</td>
</tr>
<tr>
<td>ODBC server classes</td>
<td>version 11.0.0</td>
<td>enhancement, 230</td>
</tr>
<tr>
<td>OEM.ini</td>
<td>version 11.0.0</td>
<td>enhancement, 259</td>
</tr>
<tr>
<td>offline transaction logs</td>
<td>version 16.0</td>
<td>behavior change, SQL Remote, 72</td>
</tr>
<tr>
<td>offline transaction logs</td>
<td>version 16.0</td>
<td>enhancement, MobiLink, 3</td>
</tr>
<tr>
<td>OFFSET clause</td>
<td>version 12.0.1</td>
<td>enhancement, 87</td>
</tr>
<tr>
<td>OLE DB</td>
<td>version 11.0.1</td>
<td>enhancement, 191</td>
</tr>
<tr>
<td>ON clause</td>
<td>version 16.0</td>
<td>unsupported, 39</td>
</tr>
<tr>
<td>OnDemand Edition support</td>
<td>version 12.0.1</td>
<td>new feature, 73</td>
</tr>
<tr>
<td>Open Client</td>
<td>version 11.0.0</td>
<td>enhancement, 211</td>
</tr>
<tr>
<td>OpenSSL</td>
<td>version 16.0</td>
<td>behavior changes, 3</td>
</tr>
<tr>
<td>OpenString algorithm</td>
<td>version 11.0.0</td>
<td>new feature, 209</td>
</tr>
<tr>
<td>OPENSTRING clause</td>
<td>version 11.0.0</td>
<td>new feature, 209</td>
</tr>
<tr>
<td>OpenTableEx function [UL C++]</td>
<td>version 11.0.0</td>
<td>enhancement, UltraLite, 254</td>
</tr>
<tr>
<td>OPENXML operator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>version 12.0.0 behavior changes, 144</td>
<td></td>
<td></td>
</tr>
<tr>
<td>version 12.0.0 enhancement, 128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>operator precedence, full text search</td>
<td>version 11.0.1</td>
<td>behavior change, 193</td>
</tr>
<tr>
<td>optimistic_wait_for_commit option</td>
<td>version 11.0.0</td>
<td>unsupported, 241</td>
</tr>
<tr>
<td>optimizer</td>
<td>version 11.0.0</td>
<td>enhancement, 206</td>
</tr>
<tr>
<td>version 11.0.1 enhancement, 191</td>
<td></td>
<td></td>
</tr>
<tr>
<td>version 12.0.1 enhancement, 89</td>
<td></td>
<td></td>
</tr>
<tr>
<td>version 16.0 enhancement, 14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPTION clause</td>
<td>version 11.0.0</td>
<td>behavior change, 238</td>
</tr>
<tr>
<td>version 11.0.0 enhancement, 223</td>
<td></td>
<td></td>
</tr>
<tr>
<td>options watch list</td>
<td>version 11.0.0</td>
<td>new feature, 218</td>
</tr>
<tr>
<td>options window</td>
<td>version 16.0</td>
<td>enhancement, 69</td>
</tr>
<tr>
<td>OptionWatchAction property</td>
<td>version 11.0.0</td>
<td>new feature, 215</td>
</tr>
<tr>
<td>OptionWatchList property</td>
<td>version 11.0.0</td>
<td>new feature, 215</td>
</tr>
<tr>
<td>OR REPLACE clause</td>
<td>version 11.0.1</td>
<td>enhancement, 190</td>
</tr>
<tr>
<td>version 12.0.0 new feature, 132</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td>version 11.0.0</td>
<td>enhancement, 244</td>
</tr>
<tr>
<td>version 12.0.0 enhancement, 150</td>
<td></td>
<td></td>
</tr>
<tr>
<td>version 16.0 new feature, MobiLink, 44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle driver</td>
<td>version 11.0.0</td>
<td>enhancement, 244</td>
</tr>
<tr>
<td>version 12.0.0 enhancement, 150</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle varray</td>
<td>version 11.0.1</td>
<td>new feature, 198</td>
</tr>
<tr>
<td>OSGi deployment bundle</td>
<td>version 12.0.1</td>
<td>enhancement, 77</td>
</tr>
<tr>
<td>OSUser property</td>
<td>version 11.0.0</td>
<td>new feature, 214</td>
</tr>
<tr>
<td>out of memory</td>
<td>version 16.0</td>
<td>behavior change, 38</td>
</tr>
<tr>
<td>Outbound Enabler</td>
<td>version 12.0.0</td>
<td>enhancement, Relay Server, 157</td>
</tr>
<tr>
<td>OUTER APPLY clause</td>
<td>version 11.0.0</td>
<td>new feature, 222</td>
</tr>
<tr>
<td>outer joins</td>
<td>version 12.0.0</td>
<td>enhancement, 121</td>
</tr>
<tr>
<td>outer references</td>
<td>version 12.0.0</td>
<td>enhancement, 121</td>
</tr>
</tbody>
</table>

Copyright © 2014, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0 349
aggregate functions, 298
OUTPUT statement
  version 11.0.0 behavior change, 262
  version 11.0.0 behavior change, dbisql, 240
  version 11.0.0 enhancement, 262
  version 11.0.1 behavior change, 196
  version 12.0.0 enhancement, 182
overflow errors
  version 12.0.0 enhancement, 139
overview pane
  version 11.0.0 new feature, 257
overview tab
  version 12.0.0 enhancement, 178

P

packet size
  version 11.0.0 default change, 229
PacketsReceived property
  version 12.0.1 new feature, 90
PacketsReceivedUncomp property
  version 12.0.1 new feature, 90
PacketsSent property
  version 12.0.1 new feature, 90
PacketsSentUncomp property
  version 12.0.1 new feature, 90
page sizes
  version 12.0.0 enhancement, 124
Palm OS
  version 12, deprecated feature, UltraLite, 254
parallel backups
  version 11.0.1 enhancement, 193
ParentConnection property
  version 12.0.0 new feature, 126
partial index scan
  version 12.0.1 enhancement, 89
PartnerState property
  version 12.0.1 enhancement, 85
  version 16.0 behavior change, 17
passwords
  version 11.0.0 enhancement, 209
  version 12.0.1 enhancement, 75
  version 16.0 behavior change, 33
  version 16.0 enhancement, 18
pause_after_failure control parameter
  version 12.0.1 enhancement, 101
PBUF connection parameter
  version 11.0.0 enhancement, 209
version 12.0.0 behavior change, 143
percent_as_comment option
  version 11.0.0 behavior change, 241
performance
  upgrading database files, 280
  version 12.0.0 enhancement, 138
  version 12.0.1 enhancements, 88
  version 16.0 enhancement, 30
performance data
  version 16.0 enhancement, 67
Performance Monitor
  version 11.0.0 enhancement, 214
performance statistics utility (dbstats)
  version 12.0.0 new feature, 137
performance warnings
  version 12.0.0 enhancement, 185
permissions
  version 16.0 behavior change, replaced by privileges, 7
personal server
  version 12.0.0 behavior change, 141
PHP
  version 12.0.0 enhancement, 136
  version 16.0 enhancement, 29
  version 16.0 unsupported functions, 40
PHP module
  version 11.0.0 behavior change, function name changes, 234
PID files
  version 12.0.0 new feature, 138
plan caching
  version 11.0.0 enhancement, 229
  version 16.0 behavior change, 34
plan viewer
  version 11.0.0 deprecated, features, 260
  version 11.0.0 enhancement, 260
plans
  version 12.0.1 enhancement, 76
platforms
  version 11.0.0 enhancement, UltraLite, 252
  version 12.0.0 enhancement, UltraLite, 163
  version 12.0.1 enhancement, UltraLite, 103
  version 16.0 enhancement, UltraLite, 60
plug-ins
  supported versions, 267
Pocket PC 2003
  version 12.0.1 new feature, 82
  version 16.0 enhancement, UltraLite, 60
port protocol option
  version 12.0.0 behavior change, 142
post_login_procedure option
  version 11.0.0 behavior change, 237
power failures
  version 12.0.0 enhancement, 140
precautions
  upgrading, 274
precision option
  version 11.0.0 behavior change, 237
prefetch
  version 11.0.0 enhancement, 209
PrefetchBuffer (PBUF) connection parameter
  version 12.0.0 behavior change, 143
PrefetchBuffer connection parameter
  version 11.0.0 enhancement, 209
PREFILTER clause
  version 12.0.0 new feature, 121
PREPARE statement
  version 12.0.0 behavior change, 143
Prepares property
  version 11.0.1 new feature, 188
PrepStmt property
  version 12.0.1 new feature, 90
PreserveSource property
  version 11.0.0 unsupported, 240
printing
  version 11.0.0 enhancement, Interactive SQL, 260
priority option
  version 11.0.0 new feature, 213
priority property
  version 11.0.0 new feature, 214
privileges
  version 16.0 new feature, 7
ProcessID property
  version 12.0.1 new feature, 76
ProcessorAffinity property
  version 16.0 new feature, 24
PROFILE authority
  version 16.0 deprecated, 40
progress messages
  version 12.0.0 enhancement, 128
Progress property
  version 12.0.0 new feature, 126
progress_messages option
  version 12.0.0 new feature, 125
progress_messages property
  version 12.0.0 new feature, 126
promotion_time option
  version 16.0 new feature, 17
properties windows
  version 11.0.0 deprecated, UltraLite features, 260
PROPERTY_NAME function
  version 11.0.0 enhancement, 216
proximity searches
  version 11.0.1 behavior change, 193
  version 16.0 enhancement, 31
proxy tables
  version 12.0.0 enhancement, 138
proxy_host option
  version 12.0.1 new feature, Relay Server, 98
proxy_port option
  version 12.0.1 new feature, Relay Server, 98
publication synchronization profile
  version 12.0.0 deprecated, MobiLink, 161
Python
  version 11.0.0 new feature, Python Database API, 223
  version 12.0.0 new feature, 136
Q
QAnywhere
  not supported for version 16.0, 54
queries
  version 16.0 enhancement, 14
query optimization
  version 11.0.1 enhancement, 191
  version 12.0.1 enhancement, 76
query performance
  version 11.0.1 enhancement, 191
query_mem_timeout option
  version 11.0.0 new feature, 213
query_mem_timeout property
  version 11.0.0 new feature, 214
query_plan_on_open option
  version 11.0.0 unsupported, 241
QueryBypassedCosted property
  version 11.0.1 new feature, connection property, 187
  version 11.0.1 new feature, database property, 188
QueryBypassedHeuristic property
  version 11.0.1 new feature, connection property, 187
  version 11.0.1 new feature, database property, 188
QueryBypassedOptimized property
version 11.0.1 new feature, connection property, 187
version 11.0.1 new feature, database property, 188
QueryDescribedBypass property
version 11.0.1 new feature, connection property, 187
version 11.0.1 new feature, database property, 188
QueryDescribedOptimizer property
version 11.0.1 new feature, connection property, 187
version 11.0.1 new feature, database property, 188
QueryHeapPages property
version 12.0.1 new feature, 90
querying
version 11.0.0 new feature, BLOBs, 209
version 11.0.0 new feature, files, 209
QueryMemActiveCurr property
version 11.0.0 new feature, 214
version 12.0.1 new feature, 90
QueryMemActiveEst property
version 11.0.0 new feature, 215
QueryMemActiveMax property
version 11.0.0 new feature, 215
QueryMemExtraAvail property
version 11.0.0 new feature, 214
QueryMemGrantBase property
version 11.0.0 new feature, 215
QueryMemGrantBaseMI property
version 11.0.0 new feature, 215
QueryMemGrantExtra property
version 11.0.0 new feature, 215
QueryMemGrantFailed property
version 11.0.0 new feature, 214
version 12.0.1 new feature, 90
QueryMemGrantGranted property
version 11.0.0 new feature, 214
version 12.0.1 new feature, 90
QueryMemGrantRequested property
version 11.0.0 new feature, 214
version 12.0.1 new feature, 90
QueryMemGrantWaited property
version 12.0.1 new feature, 90
QueryMemGrantWaiting property
version 11.0.0 new feature, 214
QueryMemPages property
version 11.0.0 new feature, 215
QueryMemPercentOfCache property
version 11.0.0 new feature, 215
QueryMemWaited property
version 11.0.0 new feature, 214
QueryOpened property
version 11.0.1 new feature, connection property, 187
version 11.0.1 new feature, database property, 188
QueryReused property
version 11.0.0 new feature, 216
QueryRowsBufferFetch property
version 12.0.0 unsupported, 146
QueryRowsFetched property
version 12.0.0 new feature, 126
question marks
use in SQL scripts, version 12.0.0 deprecated, 162
quick setup for Apache on Linux
version 16.0 new feature, Relay Server, 54
quick setup on IIS 7, 7.5, and 8.0
version 16.0 new feature, Relay Server, 54
quick start
upgrading databases to version 16, 275
quoted identifiers
version 11.0.0 behavior change, 237
quoted_identifier option
version 11.0.0 behavior change, 237

R
RAISERROR clause
version 12.0.1 enhancement, 87
RAISERROR statement
version 12.0.1 enhancement, 87
raw encryption
version 16.0 new feature, 19
read locks
version 12.0.0 enhancement, 138
read only
version 11.0.0 new feature, mirror database access, 209
read only databases
version 12.0.1 enhancement, 74
read only scale out
version 12.0.0 new feature, 118
READ statement
version 12.0.0 enhancement, 109
read-only database
version 11.0.1 behavior change, 202
READ_CLIENT_FILE function
version 11.0.0 new feature, 216
readcert utility
  version 11.0.0 removed, 265
READCLIENTFILE authority
  version 16.0 deprecated, 40
READFILE authority
  version 16.0 deprecated, 40
ReadHint property
  version 11.0.0 new feature, 214
  version 11.0.0 renamed, 239
ReadHintScatter property
  version 11.0.0 new feature, 214
  version 11.0.0 renamed, 239
ReadHintScatterLimit property
  version 11.0.0 new feature, 215
  version 11.0.0 renamed, 239
ReadPK locks
  version 12.0.0 new feature, 138
rebuild utility
  version 12.0.0 unsupported, 146
rebuilding
  databases for version 16, 275, 286
  precautions for upgrading, 274
  restrictions, 280
  troubleshooting, 282
  version 9 and earlier databases, 283
  version 9 and earlier databases using dbunload, 285
rebuilding databases
  failed rebuilds, 282
  version 11.0.0 enhancement, 209
rebuilding version 9 and earlier databases
  about, 280
ReceiveBufferSize protocol option
  version 12.0.1 enhancement, 75
Redirector
  version 11.0.0 deprecated, 246
redirector
  version 12.0.0 removed, 161
REFRESH MATERIALIZED VIEW statement
  version 11.0.0 behavior change, 234
  version 11.0.0 enhancement, 208
REFRESH TEXT INDEX statement
  version 11.0.0 new feature, 221
refreshing
  version 11.0.0 behavior change, materialized views, 236
  version 16.0 enhancement, MobiLink plug-in, 50
REGBIN protocol option
  version 11.0.0 unsupported, 241
REGEXP search condition
  version 11.0.0 new feature, 206
  version 11.0.1 behavior change, 194
REGEXP_SUBSTR function
  version 11.0.0 new feature, 206, 217
  version 11.0.1 behavior change, 194
RegisterBindery protocol option
  version 11.0.0 unsupported, 241
regular expressions
  version 11.0.0 new feature, 206
  version 11.0.1 behavior change, 194
relative paths
  version 16.0 behavior change, 33
Relay Server
  version 12.0.1 new features, 98
  version 16.0, 32-bit version discontinued (Linux), 57
  version 16.0, 32-bit version discontinued (Windows), 57
Relay Server behavior changes
  memory management, improved, 56
  proxy support using passthrough, 55
  quick setup for Apache on Linux, 54
  quick setup on IIS 7, 7.5, and 8.0, 54
  renew overlapped cookie, 55
  up_pad_size, 55
  version 16.0, 55
  virtual memory management, improved, 55
Relay Server enhancements
  affinity flag extended, 3
Relay Server new features
  identity forwarding, 83
  version 16.0, 54
Relay Server plug-in
  version 12.0.0 new feature, MobiLink, 149
Relay Server plug-in behavior changes
  Advanced tab, backend server farm properties, 56
  Network tab, 56
  Network tab, relay server farm properties, 56
Relay Server status page improved
  version 16.0 behavior change, 56
reload.sql
  version 11.0.0 enhancement, 230
remote data access
  version 11.0.0 behavior change, 240
  version 12.0.0 enhancement, 138
  version 12.0.1 enhancement, 86
remote data access class

Copyright © 2014, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0 353
Index

version 16.0 enhancement, 17
remote databases
  upgrading, 308
  version 12.0.0 enhancement, 130
REMOTE DBA authority
  version 16.0 deprecated, 40
remote procedures
  version 12.0.1 new feature, 86
remote schema name
  version 12.0.1 new feature, MobiLink, 97
remote table owners
  version 16.0 changes, MobiLink plug-in, 52
remote task
  version 12.0.1 new feature, MobiLink, 95
  version 16.0 enhancement, MobiLink plug-in, 50
remote_id system parameter
  version 12.0.0 new feature, MobiLink, 152
RemoteCapability property
  version 11.0.0 new feature, 215
renew overlapped cookie
  Relay Server, 55
REORGANIZE TABLE statement
  version 11.0.0 behavior changes, 234
replicate_all option
  version 12.0.0 unsupported, 114
Replication Server
  version 12.0.0 unsupported, 113
ReqCountActive property
  version 12.0.1 new feature, 90
ReqCountBlockContention property
  version 12.0.1 new feature, 90
ReqCountBlockIO property
  version 12.0.1 new feature, 90
ReqCountBlockLock property
  version 12.0.1 new feature, 90
ReqCountUnscheduled property
  version 12.0.1 new feature, 90
ReqTimeActive property
  version 12.0.1 new feature, 90
ReqTimeBlockContention property
  version 12.0.1 new feature, 90
ReqTimeBlockIO property
  version 12.0.1 new feature, 90
ReqTimeBlockLock property
  version 12.0.1 new feature, 90
ReqTimeUnscheduled property
  version 12.0.1 new feature, 90
reqtool utility
  version 11.0.0 removed, 265
request log
  version 16.0 behavior change, 34
request logging
  version 16.0 enhancement, 71
RequestsReceived property
  version 12.0.1 new feature, 90
reserved words
  troubleshooting version 16 upgrades, 282
  version 12.0.0 behavior change, 145
  version 12.0.0 enhancement, 128
  version 16.0 enhancement, 31
reserved words option
  version 12.0.0 new feature, 125
reserved_keywords option
  version 12.0.0 new feature, 125
  version 12.0.1 behavior change, 92
reset.sql
  reloading tables with autoincrement columns, 277
RESOURCE authority
  version 16.0 deprecated, 40
RESTART clause
  version 16.0 new feature, 18
restartable downloads
  version 16.0 enhancement, MobiLink, 3
restartable downloads on Android
  version 16.0 enhancement, UltraLiteJ, 64
restartable HTTP
  version 12.0.1 new feature, 81
RESTORE DATABASE statement
  version 12.0.0 enhancement, 123
  version 16.0 enhancement, 28
result sets
  version 11.0.0 new feature, making read-only, 258
  version 16.0 enhancement, 69
results
  version 11.0.0 new feature, making read-only, 258
results pane
  version 11.0.0 enhancement, 261
  version 12.0.0 enhancement, 180
ResultSet.afterLast method [UltraLiteJ]
  version 12.0.1 new feature, 78
ResultSet.beforeFirst method [UltraLiteJ]
  version 12.0.1 new feature, 78
ResultSet.first method [UltraLiteJ]
  version 12.0.1 new feature, 78
ResultSet.getBlobInputStream method [UltraLiteJ]
  version 12.0.1 enhancement, 79
ResultSet.getBoolean method [UltraLiteJ]
version 12.0.1 enhancement, 79
ResultSet.getBytes method [UltraLiteJ]
version 12.0.1 enhancement, 79
ResultSet.getClobReader method [UltraLiteJ]
version 12.0.1 enhancement, 79
ResultSet.getDate method [UltraLiteJ]
version 12.0.1 enhancement, 79
ResultSet.getDecimalNumber method [UltraLiteJ]
version 12.0.1 enhancement, 79
ResultSet.getDouble method [UltraLiteJ]
version 12.0.1 enhancement, 79
ResultSet.getFloat method [UltraLiteJ]
version 12.0.1 enhancement, 79
ResultSet.getLong method [UltraLiteJ]
version 12.0.1 enhancement, 79
ResultSet.getRowCount method [UltraLiteJ]
version 12.0.1 new feature, 78
ResultSet.getSize method [UltraLiteJ]
version 12.0.1 enhancement, 79
ResultSet.getString method [UltraLiteJ]
version 12.0.1 enhancement, 79, 81
ResultSet.getUUIDValue method [UltraLiteJ]
version 12.0.1 enhancement, 79
ResultSet.isNull method [UltraLiteJ]
version 12.0.1 enhancement, 79
ResultSet.last method [UltraLiteJ]
version 12.0.1 new feature, 78
ResultSet.relative method [UltraLiteJ]
version 12.0.1 new feature, 78
ResultSetMetaData.getAliasName method [UltraLiteJ]
version 12.0.1 enhancement, 80
ResultSetMetaData.getCorrelationName method [UltraLiteJ]
version 12.0.1 enhancement, 80
ResultSetMetaData.getDomainName method [UltraLiteJ]
version 12.0.1 enhancement, 80
ResultSetMetaData.getQualifiedName method [UltraLiteJ]
version 12.0.1 enhancement, 80
ResultSetMetaData.getTableColumnName method [UltraLiteJ]
version 12.0.1 enhancement, 80
ResultSetMetaData.getTableName method [UltraLiteJ]
version 12.0.1 enhancement, 80
ResultSetMetaData.getWrittenName method [UltraLiteJ]
version 12.0.1 enhancement, 80
RESUME statement
version 12.0.0 enhancement, 135
return codes
version 16.0 behavior change, 37
REVOKE PUBLISH statement
version 16.0 behavior change, 10
REVOKE ROLE statement
version 16.0 new feature, 9
REVOKE statement
version 12.0.0 enhancement, 119
version 16.0 enhancement, 9
REWRITE function
version 16.0 behavior change, 34
ri_trigger_time option
version 11.0.0 unsupported, 241
RIM
version 11.0.0 new feature, UltraLite support on BlackBerry, 254
Rlbk property
version 12.0.1 new feature, 90
roles
version 16.0 new feature, 7
version 16.0 new feature, MobiLink client, 46
rollback log
version 11.0.0 behavior change, 238
ROLLBACK statement
version 12.0.0 enhancement, 180
root login policy
version 16.0 enhancement, 11
root_auto_lock_time option
version 16.0 new feature, 7
row caching
version 16.0 enhancement, UltraLite, 60
ROW constructor
version 16.0 new feature, 15
row counts
version 12.0.0 enhancement, 135
row limiting
version 12.0.1 enhancement, 81
ROW LOG clause
version 16.0 new feature, 37
row reports
version 16.0 enhancement, UltraLiteJ, 62
row traversal
version 11.0.0 enhancement, UltraLite, 254
RSA TLS and HTTPS for Pocket PC 2003
  version 16.0 enhancement, UltraLite, 60
rsor
  version 12.0.0 enhancement, Relay Server, 156
Ruby
  version 11.0.1 new feature, 191

S
sa_audit_string system procedure
  version 16.0, behavior change, 32
sa_certificate_info system procedure
  version 16.0 new feature, 24
sa_char_terms system procedure
  version 11.0.0 new feature, 217
sa_clean_database system procedure
  version 12.0.1 behavior change, 89
version 16.0, behavior change, 32
sa_column_stats system procedure
  version 16.0, behavior change, 32
sa_config.csh
  version 12.0.1 enhancement, Unix, 86
sa_config.sh
  version 12.0.1 enhancement, Unix, 86
sa_conn_activity system procedure
  version 16.0 behavior change, 36
version 16.0, behavior change, 32
sa_conn_compression_info system procedure
  version 16.0 behavior change, 36
version 16.0, behavior change, 32
sa_conn_info system procedure
  version 16.0 behavior change, 36
version 16.0, behavior change, 32
sa_conn_list system procedure
  version 16.0 behavior change, 36
version 16.0, behavior change, 32
sa_conn_options system procedure
  version 16.0 behavior change, 36
version 16.0, behavior change, 32
sa_conn_properties system procedure
  version 11.0.0 enhancement, 218
version 16.0 behavior change, 36
version 16.0, behavior change, 32
sa_get_dtt_groupreads system procedure
  version 11.0.0 new feature, 216
sa_get_histogram system procedure
  version 16.0, behavior change, 32
sa_get_ldapserver_status system procedure
  version 16.0 new feature, 11
sa_get_request_profile system procedure
  version 16.0 new feature, 11
sa_get_request_times system procedure
  version 16.0, behavior change, 32
sa_get_server_messages system procedure
  version 11.0.0 deprecated, 243
sa_get_table_definition system procedure
  version 11.0.1 new feature, dbunload, 189
version 16.0, behavior change, 32
sa_get_user_status system procedure
version 11.0.0 new feature, 217
version 16.0 enhancement, 11
sa_index_density system procedure
version 11.0.0 enhancement, 218
version 12.0.0 behavior change, 144
version 16.0, behavior change, 32
sa_index_levels system procedure
version 16.0, behavior change, 32
sa_install_feature system procedure
version 12.0.0 new feature, 127
version 16.0, behavior change, 32
sa_internal_text_index_postings system procedure
version 11.0.0 new feature, for internal use only, 217
sa_java_loaded_classes system procedure
version 16.0, behavior change, 32
sa_list_cursors system procedure
version 12.0.0 new feature, 127
sa_load_cost_model system procedure
version 16.0, behavior change, 32
sa_locks system procedure
version 12.0.0 enhancement, 138
sa_make_object system procedure
version 16.0, behavior change, 32
sa_materialized_view_can_be_immediate system procedure
version 11.0.0 new feature, 218
version 16.0, behavior change, 32
sa_materialized_view_info system procedure
version 11.0.0 enhancement, 218
sa_mirror_server_status system procedure
version 12.0.0 new feature, 127
sa_nchar_terms system procedure
version 11.0.0 new feature, 217
sa_parse_json system procedure
version 16.0 new feature, 24
sa_post_login_procedure system procedure
version 11.0.0 new feature, 218
sa_procedure_profile system procedure
version 16.0, behavior change, 32
sa_procedure_profile_summary system procedure
version 16.0, behavior change, 32
sa_recompile_views system procedure
version 16.0, behavior change, 32
sa_refresh_materialized_views system procedure
version 16.0, behavior change, 32
sa_refresh_text_indexes system procedure
version 11.0.0 new feature, 217

version 16.0, behavior change, 32
sa_remove_tracing_data system procedure
version 16.0, behavior change, 32
sa_reserved_words system procedure
version 12.0.0 new feature, 128
sa_reset_identity system procedure
reloading tables with autoincrement columns, 277
version 16.0, behavior change, 32
sa_save_trace_data system procedure
version 16.0, behavior change, 32
sa_send_udp system procedure
version 16.0, behavior change, 32
sa_server_messages system procedure
version 11.0.0 new feature, 243
sa_server_option system procedure
version 11.0.0 enhancement, 218, 227
version 12.0.0 enhancement, 119, 128
version 16.0 enhancement, 21, 24
version 16.0, behavior change, 32
sa_set_http_option system procedure
version 11.0.1 enhancement, 189
sa_set_tracing_level system procedure
version 16.0, behavior change, 32
sa_table_fragmentation system procedure
version 16.0, behavior change, 32
sa_table_page_usage system procedure
version 12.0.0 enhancement, 128
version 16.0, behavior change, 32
sa_table_stats system procedure
version 16.0, behavior change, 32
sa_text_index_handles system procedure
version 11.0.0 new feature, for internal use only, 217
sa_text_index_postings system procedure
version 11.0.0 new feature, for internal use only, 217
sa_text_index_stats system procedure
version 11.0.0 new feature, 217
sa_text_index_vocab system procedure
version 11.0.0 new feature, 217
version 12.0.0 behavior changes, 144
sa_text_index_vocab_nchar system procedure
version 12.0.0 new feature, 127
version 16.0, behavior change, 32
sa_unload_cost_model system procedure
version 16.0, behavior change, 32
sa_user_defined_counter_add system procedure
version 12.0.1 new feature, 85
Index

version 16.0, behavior change, 32
sa_user_defined_counter_set system procedure
    version 12.0.1 new feature, 86
    version 16.0, behavior change, 32
sa_validate system procedure
    version 11.0.0 behavior changes, 234
    version 16.0, behavior change, 32
sa_verify_password system procedure
    version 16.0 behavior change, 35
    version 16.0, behavior change, 32
SAJDBC server class
    version 12.0.0 deprecated, 146
    version 16.0 unsupported, 41
sajdbc.jar
    version 12.0.1 unsupported, 94
    version 16.0 unsupported, 41
sajdbc4.jar
    version 12.0.1 behavior change, 94
sample certificates
    version 16.0 behavior change, 12, 52
sample database
    version 11.0.0 enhancement, SQL Anywhere , 265
    version 12.0.0 enhancement, 139
sample_config.csh
    version 12.0.1 enhancement, Unix, 86
sample_config.sh
    version 12.0.1 enhancement, Unix, 86
SAP HANA
    version 16.0 new feature, 17
    version 16.0 new feature, MobiLink, 43
SAP Sybase IQ
    version 12.0.0 new feature, 108
SAP Sybase IQ 15.2 support
    version 16.0 removed, MobiLink, 53
SAP Sybase IQ 15.4 support
    version 16.0 new feature, MobiLink, 43
SAP Sybase IQ 16
    version 16.0 enhancement, MobiLink, 3
SAP Sybase IQ support
    version 12.0.1 new feature, MobiLink, 94
SASpxOptionsBuilder class [SA .NET 2.0]
    version 11.0.0 unsupported, 241
SATMP environment variable
    version 11.0.0 enhancement, 228
save as ODBC data source
    version 11.0.0 new feature, 256
SAVE OPTION VALUES clause
    version 12.0.0 new feature, 130
scale option
    version 11.0.0 behavior change, 237
scale out
    version 12.0.0 new feature, 118
SCHEMATA rowset, OLE DB
    version 11.0.1 new feature, 191
script_full_path field
    version 16.0 removed, MobiLink, 53
scripted upload
    version 16.0 behavior change, MobiLink, 52
scripts
    upgrading, 309
scvwen1600.jar
    version 16.0 deprecated, 68
search conditions
    version 12.0.0 enhancement, 132
    version 12.0.0 new feature, 130
SearchBindery protocol option
    version 11.0.0 unsupported, 241
secure features
    version 11.0.0 behavior change, 243
    version 16.0 behavior change, 34, 68
    version 16.0 enhancement, 19, 20
secure synchronization
    version 12.0.1 enhancement, 81
secure_feature_key option
    version 16.0 deprecated, 40
security
    version 11.0.0 enhancement, UltraLite, 252
select from DML
    version 12.0.0 new feature, 121
SELECT statement
    version 11.0.0 enhancement, 206, 222
    version 11.0.1 enhancement, 190
    version 12.0.0 enhancement, 121, 132
    version 12.0.1 enhancement, 87
    version 16.0 enhancement, 9
selectivity estimates
    version 12.0.0 enhancement, 139
SELinux policies
    version 11.0.0 new feature, 228
send column names
    version 16.0 behavior change, 14
    version 16.0 behavior change, MobiLink, 51
    version 16.0 behavior change, UltraLite, 66
send_column_names synchronization parameter
    version 16.0 behavior change, UltraLite, 14
SendBufferSize protocol option
version 12.0.1 enhancement, 75
SendColumnNames extended option
version 16.0 behavior change, MobiLink, 14
version 16.0 deprecated, MobiLink, 51
sequences
version 12.0.0 new feature, 119
Server connection parameter
version 12.0.0 behavior change, 142
server farm
version 12.0.0 enhancement, MobiLink, 149
server licensing utility (dblic)
version 12.0.0 enhancement, 124
version 12.0.1 enhancement, 93
server messages
version 11.0.0 enhancement, 230
server messages window
version 12.0.0 enhancement, 139
server properties
version 11.0.0 new feature, 215
server side backups
version 11.0.1 enhancement, 193
server-initiated synchronization in a server farm
version 11.0.0 enhancement, 247
server-initiated synchronization listener
version 11.0.0 enhancement, 247
ServerEdition property
version 11.0.1 new feature, 188
ServerName connection parameter
version 12.0.0 behavior change, 142, 147
version 12.0.1 behavior change, OnDemand, 74
ServerNodeAddress property
version 11.0.0 new feature, 214
ServerPort protocol option
version 12.0.0 behavior change, 142
service utility
version 12.0.0 behavior change, 113
service utility (dbsvc)
version 11.0.1 enhancement, 188
version 12.0.0 behavior change, 147
version 12.0.0 enhancement, 124
services
version 11.0.1 enhancement, 188
version 12.0.0 behavior change, 113
version 12.0.0 enhancement, 124, 138
SET MIRROR OPTION statement
version 12.0.0 new feature, 118
version 16.0 enhancement, 17
SET OPTION statement
version 11.0.0 behavior change, Interactive SQL, 263
version 12.0.0 behavior change, 147
version 12.0.0 behavior change, Interactive SQL , 182
version 12.0.1 enhancement, 132
version 12.0.1 behavior change, UltraLite, 107
SET REMOTE OPTION
version 12.0.1 behavior change, 77
SET REMOTE OPTION statement
version 12.0.1 enhancement, 101
SET TEMPORARY OPTION statement
version 16.0 behavior change, 35
setAutoCheckpoint method
version 16.0 removed API, UltraLite , 65
setClob
PreparedStatement, 88
setDatabaseKey method
version 16.0 behavior change, UltraLite , 66
setSendColumnNames method
version 16.0 behavior change, UltraLite, 14
SetupVSPackage
version 12.0.1 enhancement, 77
shadow paging
version 16.0 removed feature, UltraLite , 65
shadow table index
version 12.0.0 new feature, MobiLink, 148
SHAPEFILE clause
version 12.0.0 new feature, FROM clause , 115
version 12.0.0 new feature, LOAD TABLE statement clause , 115
version 12.0.1 new feature, INPUT statement, 84
shapefiles
version 12.0.1 enhancement, 84
version 12.0.1, loading enhancements, 83
shared memory
version 12.0.0 behavior change, 142
version 12.0.0 enhancement, Relay Server, 157
SHARED_DIR environment variable
version 11.0.0 unsupported, 242
shortcuts
version 11.0.0 new feature, 259
showPerformanceDataUI
version 16.0 new feature, 67
Sierra Wireless Aircards
version 11.0.0 unsupported, 248
silent install
version 12.0.0 new feature, 136
silent installs
version 11.0.0 behavior change, 242
SIMILAR TO search condition
version 11.0.0 new feature, 206
version 11.0.1 behavior change, 194
single application access
version 16.0 new feature, UltraLite, 59
single step
version 11.0.0 new feature, 258
SMALLINT UNSIGNED data type
version 16.0 new feature, 28
smartphone
version 11.0.0 enhancement, 265
version 11.0.0 new feature, UltraLite, 254
version 12.0.1 enhancement, UltraLite, 103
SMS listener
version 16.0 behavior change, 51
SMTP
version 12.0.0 enhancement, 128
version 16.0 behavior change, 37
SNMP
version 11.0.0 enhancement, 231
software compatibility
UltraLite upgrades, 311
sp_alter_secure_feature_key system procedure
version 16.0 behavior change, 35
version 16.0 new feature, 19
sp_auth_sys_role_info system procedure
version 16.0 new feature, 8
sp_copy_directory system procedure
version 16.0 new feature, 25
version 16.0, behavior change, 32
sp_copy_file system procedure
version 16.0 new feature, 25
version 16.0, behavior change, 32
sp_create_directory system procedure
version 16.0 new feature, 25
version 16.0, behavior change, 32
sp_create_secure_feature_key system procedure
version 16.0 behavior change, 35
version 16.0 new feature, 19
sp_delete_directory system procedure
version 16.0 new feature, 25
version 16.0, behavior change, 32
sp_delete_file system procedure
version 16.0 new feature, 25
version 16.0, behavior change, 32
sp_displayroles system procedure
version 16.0 new feature, 8
sp_drop_secure_feature_key system procedure
version 16.0 new feature, 20
sp_forward_to_remote_server system procedure
version 12.0.1 new feature, 85
version 16.0, behavior change, 32
sp_get_last_synchronize_result system procedure
version 16.0, behavior change, 32
sp_has_role system procedure
version 16.0 new feature, 8
sp_list_directory
version 16.0 enhancement, 1
sp_list_directory system procedure
version 16.0, behavior change, 32
sp_list_secure_feature_keys system procedure
version 16.0 new feature, 20
sp_move_directory system procedure
version 16.0 new feature, 25
version 16.0, behavior change, 32
sp_move_file system procedure
version 16.0 new feature, 25
version 16.0, behavior change, 32
sp_objectpermission system procedure
version 16.0 new feature, 8
sp_password system procedure
version 16.0 behavior change, 35
sp_proc_priv system procedure
version 16.0 new feature, 8
sp_remote_columns system procedure
version 16.0, behavior change, 32
sp_remote_exported_keys system procedure
version 16.0, behavior change, 32
sp_remote_imported_keys system procedure
version 16.0, behavior change, 32
sp_remote_primary_keys system procedure
version 16.0, behavior change, 32
sp_remote_procedures system procedure
version 16.0, behavior change, 32
sp_remote_tables system procedure
version 16.0, behavior change, 32
sp_sys_priv_role_info system procedure
version 16.0 new feature, 8
sp_trace_event_fields system procedure
version 16.0 new feature, 16
sp_trace_event_session_events system procedure
version 16.0 new feature, 16
sp_trace_event_session_target_options system procedure
version 16.0 new feature, 16
sp_trace_event_session_targets system procedure
  version 16.0 new feature, 16
sp_trace_event_sessions system procedure
  version 16.0 new feature, 16
sp_trace_events system procedure
  version 16.0 new feature, 16
sp_use_secure_feature_key system procedure
  version 16.0 behavior change, 35
  version 16.0 new feature, 20
spatial data
  version 12.0.0 new feature, 115
  version 12.0.1, loading enhancements, 83
spatial data synchronization
  version 12.0.0 new feature, MobiLink, 150
spatial data types
  version 12.0.0 new feature, MobiLink, 148
Spatial Preview tab
  version 12.0.0 new feature, 115
Spatial Viewer
  version 12.0.0 new feature, 115
SpatialUtilities class
  version 12.0.0 new feature, MobiLink, 152
SPs
  database mirroring, 293
SPX
  version 11.0.0 unsupported, 241
  SpxOptionsBuilder property [SA .NET 2.0]
    version 11.0.0 unsupported, 241
  SpxOptionsString property [SA .NET 2.0]
    version 11.0.0 unsupported, 241
SQL Anywhere
  version 11.0.1 support
    version 16.0 removed, MobiLink, 53
SQL Anywhere 16
  upgrading to, 267
    version 16.0 new feature, MobiLink, 43
SQL Anywhere behavior changes
  version 11.0.0, 232
  version 11.0.1, 193
  version 12.0.0, 140
  version 12.0.1, 91
  version 16.0, 32
SQL Anywhere C API
  version 11.0.0 new feature, 223
SQL Anywhere deprecated features
  version 11.0.0, 240
  version 11.0.1, 197
  version 12.0.0, 146
  version 12.0.1, 93
  version 16.0, 39
SQL Anywhere discontinued features
  version 11.0.0, 240
  version 11.0.1, 197
  version 12.0.0, 146
  version 12.0.1, 93
  version 16.0, 39
SQL Anywhere Explorer
  version 12.0.0 unsupported, 147
SQL Anywhere for MS Access Migration utility
  version 11.0.0 unsupported, 230
SQL Anywhere JDBC driver
  version 12.0.0 new feature, 134, 135
SQL Anywhere MIB
  version 11.0.0 enhancement, 231
SQL Anywhere Monitor
  version 12.0.0 enhancement, Relay Server, 156
  version 16.0 behavior change, 70
SQL Anywhere Monitor behavior changes
  version 12.0.0, 184
  version 12.0.1, 111
SQL Anywhere new features
  version 11.0.0, 206
  version 12.0.0, 114
  version 12.0.1, 83
  version 16.0, 14
SQL Anywhere ODBC driver
  version 12.0.0 behavior change, 143
  version 12.0.0 enhancement, 134
SQL Anywhere OnDemand support
  version 12.0.1 new feature, 73
SQL Anywhere passthrough scripts
  version 11.0.0 enhancement, UltraLite, 250
SQL Anywhere plug-in
  supported versions, 267
SQL Anywhere SNMP Extension Agent
  version 11.0.0 enhancement, 231
SQL Flagger
  version 12.0.0 enhancement, 140
SQL keywords
  version 12.0.0 enhancement, 128
SQL passthrough
   version 11.0.0 new feature, 245
   version 12.0.0 discontinued, MobiLink, 160
SQL Remote
   missing messages, dealing with, 101
   upgrading, 314
   version 12.0.1 behavior change, 77
   version 16.0 new feature, 72
   version 16.0, behavior changes, 72
SQL Remote behavior changes
   version 11.0.0, 249
   version 12.0.1, 101
   version 16.0, 72
SQL Remote Message Agent (dbremote)
   version 12.0.0 behavior change, 114
SQL Remote messages
   version 12.0.1 enhancement, 101
SQL Remote new features
   version 11.0.0, 249
   version 12.0.0, 162
   version 12.0.1, 101
SQL SECURITY clause
   version 11.0.0 new feature, 217
SQL support
   version 11.0.1 enhancement, UltraLiteJ, 200
sqlanydb
   version 11.0.0 new feature, 223
   version 12.0.0 new feature, 136
SQLANYSAMP12 environment variable
   version 12.0.1 enhancement, Unix, 86
sqlanywhere_commit function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_connect function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_data_seek function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_disconnect function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_error function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_errorcode function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_execute function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_fetch_array function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_fetch_field function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_fetch_object function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_fetch_row function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_free_result function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_identity function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_num_fields function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_num_rows function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_pconnect function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_query function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_result_all function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_rollback function (deprecated)
   version 11.0.0 deprecated, 234
sqlanywhere_set_option function (deprecated)
   version 11.0.0 deprecated, 234
SQLGetConnectAttr
   version 11.0.0 behavior change, 235
SQLTables
   version 12.0.0 behavior change, 143
   version 12.0.1 behavior change, 92
   version 16.0 behavior change, 38
SRID clause
   version 12.0.1 new feature, INPUT statement, 84
SSL
   version 11.0.0 behavior change, 236
st_geometry_asbinary_format option
   version 12.0.0 new feature, 117
st_geometry_asbinary_format property
   version 12.0.0 new feature, 117
st_geometry_astext_format option
   version 12.0.0 new feature, 117
st_geometry_astext_format property
   version 12.0.0 new feature, 117
st_geometry_asxml_format option
   version 12.0.0 new feature, 117, 125
st_geometry_asxml_format property
   version 12.0.0 new feature, 117
ST_GEOMETRY_COLUMNS
   version 12.0.0 new feature, consolidated view, 116
st_geometry_describe_type option
version 12.0.1 enhancement, 78
StreamHTTPParms.getExtraParameters method
[UltraLiteJ]
    version 12.0.1 new feature, 80
StreamHTTPParms.isRestartable method [UltraLiteJ]
    version 12.0.1 new feature, 81
StreamHTTPParms.setE2eePublicKey method
[UltraLiteJ]
    version 12.0.1 enhancement, 78
StreamHTTPParms.setExtraParameters method
[UltraLiteJ]
    version 12.0.1 new feature, 80
StreamHTTPParms.setRestartable method
[UltraLiteJ]
    version 12.0.1 new feature, 81
StreamHTTPParms.setURLSuffix method
[UltraLiteJ]
    version 12.0.1 enhancement, 81
StreamHTTPParms.setZlibCompression method
[UltraLiteJ]
    version 12.0.1 enhancement, 77
StreamHTTPParms.setZlibDownloadWindowSize method [UltraLiteJ]
    version 12.0.1 enhancement, 77
StreamHTTPParms.setZlibUploadWindowSize method [UltraLiteJ]
    version 12.0.1 enhancement, 77
StreamHTTPParms.zlibCompressionEnabled method
[UltraLiteJ]
    version 12.0.1 enhancement, 77
StreamHTTPSParms.setTrustedCertificates method
[UltraLiteJ]
    version 12.0.1 enhancement, 78
StreamsUsed property
    version 11.0.0 new feature, 215
STRIP clause
    version 16.0 unsupported, 39
support packages
    version 16.0 enhancements, 1
support utility
    version 11.0.0 enhancement, 265
    version 12.0.0 behavior change, 114
support utility (dbsupport)
    version 12.0.0 enhancement, 124
    version 16.0 behavior change, 35
supported platforms
    version 11.0.0 behavior change, 238
SuppressInfoForDataTypes ODBC connection parameter
    version 16.0 new feature, 18
SUSER_NAME function
    version 16.0 behavior change, 37
SWITCHOFFSET function
    version 12.0.0 new feature, 129
Sybase Adaptive Server Enterprise 15.0 support
    version 16.0 removed, MobiLink, 53
Sybase Central
    managing older databases, 272
    rebuilding version 10 and later databases, 277
    rebuilding version 9 and earlier databases, 283
    upgrading databases, 286
    version 12.0.0 behavior change, 178
    version 16.0 behavior change, 68
    version 16.0 new feature, 67
    version support, 267
Sybase Central behavior changes
    version 11.0.0, 259
    version 11.0.1, 202
    version 12.0.0, 178
    version 16.0, 68
Sybase Central new features
    version 11.0.0, 256
    version 11.0.1, 200
    version 12.0.0, 175
    version 12.0.1, 107
    version 16.0, 67
Sybase Central plug-in
    version 16.0 new feature, 67
Sybase relay server hosting service
    version 11.0.0 new feature , 245
Sybase Replication Server
    version 12.0.0 unsupported, 113
SYNC_PROFILEOPTION_VALUE function
    version 11.0.1 enhancement, UltraLite, 199
synchronization model testing
    version 16.0 new feature, MobiLink plug-in, 48
synchronization models
    version 12.0.0 behavior change, MobiLink, 159
    version 12.0.1 new feature, MobiLink, 97
    version 16.0 enhancement, MobiLink, 3
synchronization parameters
    version 16.0 behavior change, UltraLite, 14
synchronization profile options
    version 16.0 behavior change, UltraLite, 14
synchronization profiles
version 11.0.0 enhancement, UltraLite, 250
version 11.0.0 new feature, 247
synchronization stream security
version 11.0.0 new feature, 247
SynchronizationSchemaChangeActive property
version 12.0.0 new feature, 126
SYNCHRONIZE statement
version 12.0.0 new feature, 149
SyncResult interface
version 16.0 enhancement, UltraLiteJ, 62
SYS_REPLICATION_ADMIN_ROLE system role
version 16.0 new feature, MobiLink client, 46
SYS_RUN_REPLICATION_ROLE system role
version 16.0 new feature, MobiLink client, 46
SYS_SPATIAL_ADMIN_ROLE group
version 12.0.0 new feature, 123
SYSCERTIFICATE
version 16.0 new feature, 29
SYSCOLSTAT
version 16.0 enhancement, 29
SYSDATETIMEOFFSET function
version 12.0.0 new feature, 129
SYS_EXTERNALENVOBJECT
version 16.0 enhancement, 29
SYSGROLEGRANT
version 16.0 new feature, 10
SYSGROLEGRANTEXT
version 16.0 new feature, 10
SYSGROLEGRANTS
version 16.0 new feature, 10
SYSGROUP
version 16.0 behavior change, 10
SYSGROUPS
version 16.0 deprecated, 10
SYSHISTORY
version 16.0 enhancement, 29
SYSTYPEINDEXES
version 11.0.0 behavior change, 232
SYSTABLEPERM
version 16.0 enhancement, 10
system events on Android
version 16.0 enhancement, UltraLiteJ, 64
system objects
version 16.0 enhancement, MobiLink plug-in, 50
system path
UltraLite utilities, 311
utilities, 275
SYSTEM PROCEDURE AS DEFINER clause
ALTER DATABASE, version 16.0 new feature, 33
CREATE DATABASE, version 16.0 new feature, 32
version 16.0 new feature, CREATE DATABASE
statement, 32
system procedures
version 12.0.1 behavior change, OnDemand, 74
version 16.0 behavior change, 32
system secure feature keys
  version 16.0 behavior change, 34
system tables
  version 11.0.0 behavior change, 232
  version 16.0 enhancement, 29
system views
  version 11.0.0 behavior change, 232
  version 16.0 enhancement, 29
SYSTEXTCONFIG
  version 11.0.0 new feature, 232
SYSTEXTIDX
  version 11.0.0 new feature, 232
  version 16.0 enhancement, 29
SYSTEXTIDXTAB
  version 11.0.0 new feature, 232
SYSTEXTOFMEASURE
  version 12.0.0 new feature, system view, 116
SYSUSER
  system view, version 16.0 behavior change, 39
  version 16.0 enhancement, 10, 12, 29
SYSUSERAUTHORITY
  version 16.0 behavior change, 10
SYSUSERTYPE
  version 12.0.0 enhancement, 137
SYSVIEW
  version 16.0 enhancement, 29

T
Table API
  version 11.0.0 enhancement, UltraLite, 254
table encryption
  version 11.0.0 enhancement, 210
table mapping direction
  version 12.0.1 behavior change, MobiLink, 82
table mapping editor
  version 16.0 behavior change, 49
tables, download only
  version 12.0.1 new feature, UltraLite, 102
TCP/IP
  version 12.0.0 behavior change, 141
  version 12.0.1 behavior change, 74
TcpIpAddresses property
  version 11.0.0 new feature, 215
TDS communication protocol
  version 12.0.0 enhancement, 136
temporary connections
  version 12.0.0 enhancement, 122
temporary files
  version 11.0.0 enhancement, Unix, 228
temporary functions
  version 11.0.1 enhancement, 190
temporary procedures
  version 11.0.1 enhancement, 190
temporary tables
  version 11.0.0 behavior change, 240
  version 12.0.0 enhancement, 127
TERM BREAKER clause
  version 12.0.0 new feature, 121
text completion
  version 12.0.1 enhancements, 108
  version 16.0 enhancement, 69
text configuration objects
  version 11.0.0 new feature, 206
text indexes
  version 11.0.0 new feature, 206
  version 11.0.1 behavior change, 193
  version 12.0.1 enhancement, 88
  version 16.0 enhancement, 26
text plans
  version 11.0.0 deprecated, Interactive SQL features, 260
ThreadDeadlocksAvoided property
  version 12.0.0 new feature, 127
ThreadDeadlocksReported property
  version 12.0.0 new feature, 127
threads
  version 12.0.0 behavior change, 141
Timeout protocol option
  version 12.0.1 enhancement, 75
TIMESTAMP WITH TIME ZONE data type
  version 12.0.0 new feature, 133
  version 12.0.0 new feature, MobiLink, 150
timestamp_with_time_zone_format option
  version 12.0.0 new feature, 125
TIMESTAMPADD
  version 16.0 enhancement, 2
TIMESTAMPDIFF
  version 16.0 enhancement, 2
timestamps
  version 12.0.0 enhancement, 125
TimestampWithTimeZone class
  version 12.0.0 new feature, MobiLink, 152
TimeWithoutClientConnection
  version 16.0 enhancement, 2
TINYINT UNSIGNED data type
version 16.0 new feature, 28

TLS
  version 11.0.0 behavior change, 236
  version 12.0.0 enhancement, MobiLink, 149
  version 16.0 enhancement, MobiLink, 43
TLS identities stored in database
  version 12.0.1 new feature, UltraLite, 102
TLS support
  version 16.0 behavior change, 13
tls_type argument
  ENCRYPTION connection parameter, 13
tls_type protocol option
  -ec database server option, 13
  version 16.0 deprecated, MobiLink, 54
TODATETIMEOFFSET function
  version 12.0.0 new feature, 129
toolbars
  version 11.0.0 enhancement, Interactive SQL, 261
TOP clause
  version 12.0.1 enhancement, 87
total_download_row_count
  version 16.0 enhancement, UltraLite C/C++, 61
TRUNCATE privilege
  version 16.0 new feature, 7
TRUNCATE TEXT INDEX statement
  version 11.0.0 new feature, 221
truncate_with_auto_commit option
  version 11.0.0 unsupported, 241
trusted certificates
  version 12.0.1 enhancement, 78
trusted_certificate_name options
  version 16.0 new feature, 46
trusted_certificates option
  version 16.0 new feature, MobiLink, 44
trusted_certificates_file option
  version 16.0 new feature, 12
trusted_certificates_file property
  version 16.0 new feature, 24
TRY statement
  version 16.0 new feature, 27
tsql_hex_constant option
  version 11.0.0 unsupported, 241

U

UCA collation
  version 12.0.0 enhancement, 139
UDP listener
  version 16.0 behavior change, 51
ul_column_name_type enumeration [UltraLite C++]
  version 12.0.1 new feature, 81
ul_stream_error struct
  version 11.0.0 enhancement, UltraLite, 252
ul_sync_info
  version 11.0.0 behavior change, UltraLite, 255
ul_sync_stats_download
  version 16.0 enhancement, UltraLite C/C++, 61
ul_sync_stats_upload
  version 16.0 enhancement, UltraLite C/C++, 61
ulcpp11.cpp
  version 16.0 removed API, UltraLite, 66
uleng12 utility
  version 12.0.0 behavior change, 114
uliface.h
  version 16.0 removed API, UltraLite, 66
ULjDbT
  version 11.0.1 UltraLiteJ utility, 200
ULjException class [UltraLiteJ]
  version 12.0.1 enhancement, 80
ULODBC server class
Index

version 11.0.0 new feature, 230
ULResultSetSchema.GetColumnName method
[UltraLite C++]
version 12.0.1 enhancement, 81
UltraLite
error messages, new, 107
upgrading databases and applications, 311
UltraLite ALTER DATABASE SCHEMA FROM FILE statement
version 11.0.0 enhancement, UltraLite, 251
UltraLite ALTER SYNCHRONIZATION PROFILE statement
version 11.0.0 new feature, UltraLite, 250
UltraLite applications
version 11.0.0 new feature, UltraLite row traversal, 254
UltraLite behavior changes
version 11.0.0, 254
version 12.0.0, 167
version 12.0.1, 106
version 16.0, 65
UltraLite C/C++
version 16.0 enhancement, UltraLite, 61
UltraLite C/C++ API
version 11.0.0 enhancement, UltraLite, 254
UltraLite clients
version 11.0.0 enhancement, UltraLite, 252
UltraLite CREATE SYNCHRONIZATION PROFILE statement
version 11.0.0 enhancement, UltraLite, 250
UltraLite databases
new features in version 12.0.0, 162
new features in version 12.0.1, 101
new features in version 16.0, 57
upgrade previous versions, 311
version 11.0.0 enhancement, 250
version 11.0.0 new feature, 249
UltraLite DROP SYNCHRONIZATION PROFILE statement
version 11.0.0 enhancement, UltraLite, 250
UltraLite ESQL
version 11.0.0 enhancement, UltraLite, 254
UltraLite for M-Business Anywhere
deprecated, 104
version 11.0.0 enhancement, UltraLite, 254
version 12.0.0 enhancement, UltraLite, 166
version 16.0 removed API, UltraLite, 66
UltraLite Initialize Database utility (ulinit)
 UltraLite isolation_level option
version 11.0.0 enhancement, UltraLite, 253
UltraLite Java edition databases
new features in version 16.0, 57
UltraLite LOAD TABLE statement
version 11.0.0 enhancement, UltraLite, 252
UltraLite new features
version 11.0.0, 249
version 11.0.1, 199
version 12.0.0, 162
version 12.0.1, 101
version 16.0, 57
UltraLite SELECT statement
version 11.0.0 enhancement, UltraLite, 250
UltraLite SYNCHRONIZE statement
version 11.0.0 enhancement, UltraLite, 250
UltraLite Table API
version 11.0.0 enhancement, UltraLite, 254
UltraLite tables
version 11.0.0 new feature, UltraLite row traversal, 254
UltraLiteUnload Database to XML utility (ulunload)
version 11.0.0 enhancement, UltraLite, 253
UltraLite validate database utility (ulvalid)
version 11.0.0 new feature, UltraLite, 251
UltraLite.NET
asynchronous synchronization, 103
version 11.0.0 enhancement, UltraLite, 254
version 12.0.1 enhancement, UltraLite, 103
version 16.0 enhancement, UltraLite, 61
UltraLite_Connection object
version 11.0.0 new feature, UltraLite row traversal, 254
UltraLiteJ
code sample locations, 103
version 12.0.1 enhancement, UltraLite, 104
version 16.0 enhancement, 62
UltraLiteJ behavior changes
version 11.0.1, 199, 200
UltraLiteJ code sample locations
version 12.0.1 new feature, UltraLite, 103
UltraLiteJ database transfer utility
BlackBerry, 200
UltraLiteJ databases
new features in version 12.0.1, 101
version 12.0.0 enhancement, 174
upload row count on Android version 16.0 enhancement, UltraLiteJ, 63

upload_update
detecting conflicts, version 12.0.0 deprecated, 162

user defined functions
version 12.0.1 enhancement, 76
version 12.0.0 enhancement, 138

user_estimates option
version 11.0.0 enhancement, 223

USER_NAME function
version 16.0 behavior change, 37

UserDefinedCounterRate01 property
version 12.0.1 new feature, 89
UserDefinedCounterRate02 property
version 12.0.1 new feature, 89
UserDefinedCounterRate03 property
version 12.0.1 new feature, 89
UserDefinedCounterRate04 property
version 12.0.1 new feature, 89
UserDefinedCounterRate05 property
version 12.0.1 new feature, 89
UserDefinedCounterRaw01 property
version 12.0.1 new feature, 89
UserDefinedCounterRaw02 property
version 12.0.1 new feature, 89
UserDefinedCounterRaw03 property
version 12.0.1 new feature, 89
UserDefinedCounterRaw04 property
version 12.0.1 new feature, 89
UserDefinedCounterRaw05 property
version 12.0.1 new feature, 89

username system parameter
version 12.0.0 new feature, MobiLink, 152

UTC_TIMESTAMP special value
version 12.0.0 behavior change, 141

UTC_TIMESTAMPCatalog property
version 16.0 new feature, 24

UTF-16 encoding
version 11.0.0 enhancement, CSCONVERT function, 228
version 11.0.0 enhancement, LOAD TABLE statement, 228
version 11.0.0 enhancement, UNLOAD statement, 228

UTF-8
version 11.0.0 behavior change, support in URLs, 224

utilities
version 12.0.0 behavior change, 114
version 12.0.0 enhancement, 113
version 16.0 behavior change, UltraLite, 66

utility database
version 11.0.0 behavior change, 235
version 12.0.0 enhancement, 125

UUID.toString method [UltraLite for M-Business Anywhere]
version 12.0.1 new feature, 81

uuid_has_hyphens option
version 11.0.0 unsupported, 241
version 12.0.0 reinstated feature, 124

uuid_has_hyphens property
version 12.0.0 reinstated feature, 124

V

V_TRANSACTION Oracle system view
version 12.0.0 behavior change, MobiLink, 157

VALIDATE authority
version 16.0 deprecated, 40

VALIDATE DATABASE statement
version 12.0.0 behavior change, 144

validate database wizard
version 11.0.0 new feature, 257

VALIDATE LDAP SERVER statement
version 16.0 new feature, 12

VALIDATE statement
version 12.0.1 enhancement, 87

VALIDATE TEXT INDEX statement
version 12.0.1 new feature, 87

validation
version 11.0.0 behavior change, 212
version 12.0.0 behavior change, 144
version 12.0.0 enhancement, 138

validation utility (dbvalid)
version 11.0.0 enhancement, 212
version 12.0.0 behavior change, 144

VARCHAR data type
version 12.0.0 enhancement, 133

varray (Oracle)
version 11.0.1 new feature, 198

varray support
version 12.0.0 enhancement, 150

verify_password_function option
version 16.0 enhancement, 18
version 11.0.0
behavior changes, 205
deprecated features, 205
new features, 205
version 11.0.0 enhancement
LEN function, 228
LENGTH function, 228
version 11.0.1
behavior changes, 187
deprecated features, 187
new features, 187
version 12.0.0
behavior changes, 113
Interactive SQL new features, 175
MobiLink new features, 148
new features, 113
SQL Anywhere new features, 114
Sybase Central new features, 175
UltraLite new features, 162
version 12.0.0 behavior change
index hints, 145
version 12.0.1
behavior changes, 73
behavior changes, MobiLink, 98
behavior changes, UltraLite, 106, 107
deprecated features, MobiLink, 100
discontinued features, MobiLink, 100
enhancement, UltraLite, 103
Interactive SQL new features, 107
MobiLink new features, 94
new features, 73
SQL Anywhere new features, 83
Sybase Central new features, 107
UltraLite new features, 101
version 12.0.1 behavior changes
MobiLink client, 100
MobiLink plug-in, 100
version 12.0.1 enhancement
UltraLiteJ, 104
version 12.0.1 new features
MobiLink clients, 98
Relay Server, 98
version 16.0
behavior changes, 1
behavior changes, UltraLite, 65
Interactive SQL new features, 69
known issues for upgrading, 282
MobiLink new features, 42
new feature, SQL Remote, 72
new features, 1
Relay Server behavior changes, 55
Relay Server new features, 54
removed APIs, UltraLite, 66
removed features, UltraLite, 65
removed platforms, UltraLite, 66
SQL Anywhere new features, 14
support package enhancements, 1
Sybase Central new features, 67
UltraLite new features, 57
upgrading to, 267
version 16.0 enhancement
UltraLite C/C++, 61
UltraLite.NET, 61
UltraLiteJ, 62
version 5
upgrading SQL Remote installations, 314
versions
upgrading synchronization scripts, 309
versions 12.0.0
deprecated features, 113
versions 12.0.1
deprecated features, 73
versions 16.0
deprecated features, 1
viewcert utility
version 16.0 behavior change, 3
version 16.0 enhancement, 22
views
version 11.0.0 enhancement, 228
version 16.0 behavior change, 33, 39
VIM message type
version 11.0.0 unsupported, 249
Vista
version 11.0.0 enhancement, 209
Visual Studio
version 11.0.0 behavior change, 266
Visual Studio .NET
version 11.0.0 behavior change, 266
Visual Studio 2005 project files
version 16.0 removed feature, UltraLite, 65
Visual Studio 2012 support
version 16.0 new feature, 28
VSS
version 11.0.0 new feature, 210
W
WaitStartTime property
  version 11.0.1 new feature, connection property, 187
WaitType property
  version 11.0.1 new feature, connection property, 187
warning messages on Android
  version 16.0 enhancement, UltraLiteJ, 63
watch list
  version 11.0.0 new feature, 218
web servers
  version 11.0.0 behavior change, 236
web service clients
  version 11.0.1 behavior change, 197
web services
  version 11.0.0 behavior change, 236
  version 12.0.0 enhancement, 134
  version 12.0.1 enhancement, 88
WebClientLogFile property
  version 11.0.0 new feature, 227
WebClientLogging property
  version 11.0.0 new feature, 227
webservice_sessionid_name option
  version 16.0 new feature, 23
webservice_sessionid_name property
  version 16.0 new feature, 24
what's new in version 11.0.0
  about, 205
what's new in version 11.0.1
  about, 187
what's new in version 12.0.0
  about, 113
what's new in version 12.0.1
  about, 73
what's new in version 16.0
  about, 1
wide character data streaming
  version 16.0 enhancement, UltraLite C/C++, 61
win32 directory
  version 11.0.0 behavior change, 266
Windows
  version 11.0.0 behavior change, 238
Windows 2000
  version 12.0.0 unsupported, 146
Windows CE now Windows Mobile
  version 11.0.0 behavior change, 265
Windows IIS support
  version 12.0.0 enhancement, Relay Server, 156
Windows Mobile
  version 11.0.0 enhancement, 227
  version 12.0.0 enhancement, 137
  version 16.0 behavior change, FIPS support removed, 3
Windows Performance Monitor
  version 11.0.0 enhancement, 214
Windows services
  version 12.0.0 behavior change, 113
Windows Vista
  version 11.0.0 enhancement, 209
WITH CONTENT LOGGING clause
  version 11.0.0 new feature, 219
WITH FILE NAME LOGGING clause
  version 11.0.0 new feature, 219
WITH NULLS NOT DISTINCT clause
  version 12.0.0 new feature, 130, 177
WITH ROW LOGGING clause
  version 11.0.0 new feature, 219
WITH SHARE MODE clause, REFRESH
MATERIALIZED VIEW statement
  version 11.0.0 new feature, 208
wizards
  version 11.0.0 enhancement, 260
  version 11.0.0 new feature, 257
Word documents, full text indexing
  version 12.0.0 new feature, 121
workers
  version 12.0.0 behavior change, 141
write checksums
  version 12.0.0 new feature, 122
write-at-end persistent stores
  version 16.0 removed feature, UltraLite, 66
WRITE_CLIENT_FILE function
  version 11.0.0 new feature, 216
WriteChecksums property
  version 12.0.0 new feature, 126
WRITECLIENTFILE authority
  version 16.0 deprecated, 40
WriteNoPK locks
  version 12.0.0 new feature, 138
X
X64 directory
  version 11.0.0 behavior change, 266
Xcode 4.2 samples
version 12.0.1 new feature, 82
XMLTYPE data type
version 16.0 new feature, MobiLink, 42
version 16.0 new feature, MobiLink plug-in, 50
xp_cmdshell system procedure
version 16.0, behavior change, 32
xp_get_mail_error_code system procedure
version 16.0 new feature, 25
xp_get_mail_error_text system procedure
version 16.0 new feature, 25
xp_getenv
version 16.0 enhancement, 1
xp_getenv system procedure
version 16.0 new feature, 25
version 16.0, behavior change, 32
xp_read_file system procedure
version 12.0.0 enhancement, 128
version 16.0, behavior change, 32
xp_sendmail system procedure
version 16.0 behavior change, 37
version 16.0, behavior change, 32
xp_startmail system procedure
version 16.0 behavior change, 35, 37
version 16.0, behavior change, 32
xp_startsmtp system procedure
version 12.0.0 enhancement, 128
version 16.0 behavior change, 35, 37
version 16.0 enhancement, 25
version 16.0, behavior change, 32
xp_stopmail system procedure
version 16.0 behavior change, 37
version 16.0, behavior change, 32
xp_stopsmtpe system procedure
version 16.0 behavior change, 37
version 16.0, behavior change, 32
xp_write_file system procedure
version 16.0, behavior change, 32

Z
zlib compression
version 16.0 new feature, UltraLite, 57, 58