

A large, stylized graphic of the letter 'P' composed of multiple parallel green lines, positioned in the upper right quadrant of the page.

Progress[®] DataDirect[®] OpenAccess[™] SDK 8.1

Upgrade Notes

September 2016

Contents

OPENACCESS SDK™ VERSION 8.1 5

HIGHLIGHTS5
 OpenAccess SDK SQL Engine and IP SDK.....5
 OpenAccess SDK Server.....5
 OpenAccess SDK Client for JDBC5
 OpenAccess SDK Client for ODBC5
 OpenAccess SDK Client for ADO.NET.....5
UPGRADING FROM OPENACCESS™ SDK 8.0 TO 8.16
 Backward Compatibility6
 Documentation.....6
 License Keys6
 C/C++ IP for OpenAccess™ SDK SQL Engine6
 Java IP for OpenAccess™ SDK SQL Engine7
 .NET IP for OpenAccess™ SDK SQL Engine.....7
 C SQL IP.....8
 Java SQL IP.....8
 Branding and Distribution.....8
 File Name Changes.....9
 Configuration Changes.....9

OPENACCESS SDK™ VERSION 8.0 10

HIGHLIGHTS10
 OpenAccess SDK SQL Engine and IP SDK.....10
 OpenAccess SDK Server.....10
 OpenAccess SDK Client for JDBC10
 OpenAccess SDK Client for ODBC10
 OpenAccess SDK Client for ADO.....10
 OpenAccess SDK Client for ADO.NET.....10
UPGRADING FROM OPENACCESS™ SDK 7.2 TO 8.010
 Documentation.....11
 License Keys11
 C/C++ IP for OpenAccess™ SDK SQL Engine11
 Java IP for OpenAccess™ SDK SQL Engine12
 .NET IP for OpenAccess™ SDK SQL Engine.....13
 C SQL IP.....14
 Java SQL IP.....15
 Branding and Distribution.....15
 File Name Changes.....15
 Configuration Changes.....15

OPENACCESS SDK™ VERSION 7.2.0 16

HIGHLIGHTS16
 OpenAccess SDK SQL Engine and IP SDK.....16
 OpenAccess SDK Server.....16
 OpenAccess SDK Client for JDBC16
 OpenAccess SDK Client for ODBC16
 OpenAccess SDK Client for ADO.....16
 OpenAccess SDK Client for ADO.NET.....16
UPGRADING FROM OPENACCESS™ SDK 7.0.1 TO 7.2.017
 Documentation.....17
 License Keys17
 C/C++ IP for OpenAccess™ SDK SQL Engine17
 Java IP for OpenAccess™ SDK SQL Engine17

<i>.Net IP for OpenAccess™ SDK SQL Engine</i>	18
<i>Branding and Distribution</i>	18
<i>File Name Changes</i>	18
<i>Configuration Changes</i>	18
OPENACCESS™ SDK VERSION 7.0.1.....	19
HIGHLIGHTS	19
<i>OpenAccess™ SDK SQL Engine and IP SDK</i>	19
<i>OpenAccess™ SDK Server</i>	19
<i>OpenAccess™ SDK Client for JDBC</i>	19
<i>OpenAccess™ SDK Client for ODBC</i>	19
<i>OpenAccess™ SDK Client for ADO</i>	19
<i>OpenAccess™ SDK Client for ADO.NET</i>	19
UPGRADING FROM OPENACCESS™ SDK 7.0 TO 7.0.1.....	20
<i>Documentation</i>	20
<i>License Keys</i>	20
<i>Branding and Distribution</i>	20
<i>File Name Changes</i>	20
<i>Configuration Changes</i>	20
OPENACCESS™ SDK VERSION 7.0.....	20
HIGHLIGHTS	20
<i>OpenAccess™ SDK SQL Engine and IP SDK</i>	20
<i>OpenAccess™ SDK Server</i>	21
<i>OpenAccess™ SDK Client for JDBC</i>	21
<i>OpenAccess™ SDK Client for ODBC</i>	21
<i>OpenAccess™ SDK Client for ADO</i>	21
<i>OpenAccess™ SDK Client for ADO.NET</i>	21
UPGRADING FROM OPENACCESS™ SDK 6.0 SP3 TO 7.0	22
<i>SSL Compatibility</i>	22
<i>License Keys</i>	22
<i>C/C++ IP for OpenAccess™ SDK SQL Engine</i>	22
<i>C/C++ IP for Third-party SQL Engine</i>	23
<i>Java IP for OpenAccess™ SDK SQL Engine</i>	23
<i>Java IP for Third-party SQL Engine</i>	24
<i>.NET IP for OpenAccess™ SDK SQL Engine</i>	24
<i>Branding and Distribution</i>	25
OPENACCESS™ SDK VERSION 6.0 SP3.....	27
HIGHLIGHTS	27
UPGRADING FROM OPENACCESS™ SDK 6.0 SP2 TO 6.0 SP3	27
<i>Documentation</i>	28
<i>License Keys</i>	28
<i>C/C++ IP for OpenAccess™ SDK SQL Engine</i>	28
<i>C/C++ IP for Third-party SQL Engine</i>	28
<i>Java IP for OpenAccess™ SDK SQL Engine</i>	28
<i>Java IP for Third-party SQL Engine</i>	28
<i>.NET IP for OpenAccess™ SDK SQL Engine</i>	28
<i>Branding and Distribution</i>	28
UPGRADING FROM OPENACCESS™ SDK 6.0 SP1 TO 6.0 SP3	29
<i>Documentation</i>	29
<i>License Keys</i>	29
<i>C/C++ IP for OpenAccess™ SDK SQL Engine</i>	29
<i>C/C++ IP for Third-party SQL Engine</i>	29
<i>Java IP for OpenAccess™ SDK SQL Engine</i>	30

<i>Java IP for Third-party SQL Engine</i>	30
<i>.NET IP for OpenAccess™ SDK SQL Engine</i>	30
<i>Branding and Distribution</i>	30
OPENACCESS™ SDK VERSION 6.0 SP1.....	33
HIGHLIGHTS	33
UPGRADING FROM OPENACCESS™ SDK 6.0 TO 6.0 SP1	34
<i>Documentation</i>	34
<i>License Keys</i>	34
<i>Configuration Changes</i>	35
<i>C/C++ IP for OpenAccess™ SDK SQL Engine</i>	37
<i>Java IP for OpenAccess™ SDK SQL Engine</i>	37
<i>.NET IP for OpenAccess™ SDK SQL Engine</i>	38
<i>Branding and Distribution</i>	38
OPENACCESS SDK™ VERSION 6.0.....	39
HIGHLIGHTS	39
UPGRADING FROM OPENACCESS™ SDK 5.6 TO 6.0	39
<i>Configuration Changes</i>	40
<i>C/C++ IP for OpenAccess™ SDK SQL Engine</i>	40
<i>C/C++ IP for Third-party SQL Engine (previously referred to as SQL IP)</i>	42
<i>Java IP for OpenAccess™ SDK SQL Engine</i>	44
<i>Java IP for Third-party SQL engine (previously referred to as SQL IP)</i>	47
<i>.NET IP for OpenAccess™ SDK SQL engine</i>	48

OpenAccess SDK™ Version 8.1

Highlights

This section highlights what's new in OpenAccess SDK 8.1:

OpenAccess SDK SQL Engine and IP SDK

- Support for RIGHT OUTER JOIN operations
- Improved support for nested join queries
- Improved performance with indexed disk cache for nested query results
- Improved calculation of precision of the scalar function result, based on precision of input arguments
- Support for configurable behavior to enforce an integrity constraint check when null values are specified in non-nullable result columns with the IP_INFO_VALIDATE_NULL_CONSTRAINT ipGetinfo option

OpenAccess SDK Server

- Support for service configurable option, ServiceNetworkBufferSize, to specify network protocol buffer size
- Support for service configurable option, ServiceSSLVersions, to specify the version of cryptoprotocol to be used for encryption, when SSL is enabled
- Upgrade of OpenSSL library to 1.0.2h

OpenAccess SDK Client for JDBC

- Support for JDBC 4.0 specification
- Support for the CryptoProtocolVersion connection option that specifies the version of the SSL standard used for encryption, whenever SSL is enabled

OpenAccess SDK Client for ODBC

- Distribution of the sample Tableau TDC file with the OpenAccess ODBC client as a standard component
- Support for the CryptoProtocolVersion connection option that specifies the version of the SSL standard that is used for encryption, whenever SSL is enabled
- Upgrade of OpenSSL library to 1.0.2h

OpenAccess SDK Client for ADO.NET

- Support for 64-bit row fetch
- Support for public/global IP address for client identification
- Support for the CryptoProtocolVersion connection option that specifies the version of the SSL standard that is used for encryption, whenever SSL is enabled
- Inclusion of branding tool as part of installation

- Support for Visual Studio 2013 and 2015
- Dropped support for Visual Studio 2008 and 2010
- Support for .NET Framework 4.5 and .NET Framework 4.6
- Dropped support for .NET Framework 2.0, 3.0, 3.5, and 4.0

Upgrading from OpenAccess™ SDK 8.0 to 8.1

This section provides details for upgrading from the OpenAccess SDK 8.0 release to the 8.1 release. If you are upgrading from an earlier version, first refer to the “[Upgrading from OpenAccess™ SDK 7.2 to 8.0](#)” section of this document and then return here.

Backward Compatibility

- OpenAccess SDK 8.1 Servers are backward compatible with OpenAccess SDK Client 8.0
- OpenAccess SDK 8.1 Clients are compatible with OpenAccess SDK Server 8.0

The following sections detail the changes for each of the different parts of the product.

Documentation

New documentation is being released for OpenAccess SDK 8.1. All changes to the OpenAccess SDK Version 8.0 are described in the fixes_<component>.txt, or this document.

License Keys

C/C++ IP for OpenAccess™ SDK SQL Engine

Refer to the OpenAccess SDK SQL Engine Programmer’s Reference for C/C++ for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code to the new folder and make changes to the IP APIs and DAM/DAMEX APIs as described in the following sections.

IP API Changes

Following is the enhancement for OAIP_schema introduced in 8.1:

- The pSearchObj parameter is now available even for the DAMOBJ_TYPE_SCHEMA schema type. The IP can use this information or pass the pSearchObj to the OpenAccess SDK SQL engine when calling the dam_add_damobj_xxxxx functions. It is NULL if no search criteria is specified.

DAM API changes

- Enhanced the dam_getInfo API to retrieve the client version with the DAM_INFO_CLIENT_PRODUCT_VERSION info type.
- Enhanced the dam_getInfo API to retrieve the cipher suite and cryptoprotocol version being used to connect to the OpenAccess SDK Server with the following info types:
 - *DAM_INFO_SESSION_CIPHER_SUITE*
 - *DAM_INFO_SESSION_CRYPTOPROTOCOL_VERSION*

Java IP for OpenAccess™ SDK SQL Engine

Refer to the OpenAccess SDK SQL Engine Programmer's Reference for Java for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code to the new folder and make changes to the IP APIs and DAM/DAMEX APIs as described in the following sections.

IP API Changes

Following is the enhancement for ipschema introduced in 8.1:

- The pSearchObj parameter is now available even for the DAMOBJ_TYPE_SCHEMA schema type. The IP can use this information or pass the pSearchObj to the OpenAccess SDK SQL engine when calling the dam_add_damobj_XXXXX functions. It is NULL if no search criteria is specified.

DAM API Changes

- Enhanced the dam_getInfo API to retrieve the client version with the DAM_INFO_CLIENT_PRODUCT_VERSION info type.
- Enhanced the dam_getInfo API to retrieve the cipher suite and cryptoprotocol version being used to connect to the OpenAccess SDK Server with the following info types:
 - *DAM_INFO_SESSION_CIPHER_SUITE*
 - *DAM_INFO_SESSION_CRYPTOPROTOCOL_VERSION*

.NET IP for OpenAccess™ SDK SQL Engine

Refer to the OpenAccess SDK SQL Engine Programmer's Reference for .NET for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code to the new folder and make changes to the IP APIs and DAM/DAMEX APIs as described in the following sections

IP API Changes

Following is the enhancement for ipschema introduced in 8.1:

- The pSearchObj parameter is now available even for the DAMOBJ_TYPE_SCHEMA schema type. The IP can use this information or pass the pSearchObj to the OpenAccess SDK SQL engine when calling the dam_add_damobj_XXXXX functions. It is NULL if no search criteria is specified.

DAM API Changes

- Enhanced the dam_getInfo API to retrieve the client version with the DAM_INFO_CLIENT_PRODUCT_VERSION info type.
- Enhanced the dam_getInfo API to retrieve the cipher suite and cryptoprotocol version being used to connect to the OpenAccess SDK Server with the following info types:
 - *DAM_INFO_SESSION_CIPHER_SUITE*
 - *DAM_INFO_SESSION_CRYPTOPROTOCOL_VERSION*

C SQL IP

Refer to the OpenAccess™ SDK Third-Party SQL Engine Programmer's Guide for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code to the new folder and make changes to the IP APIs as described below.

IP API Changes

- Enhanced the `OASQLIP_set_connect_info` API to retrieve the client type and client version with the following info types:
 - `OADS_CONN_INFO_CLIENT_TYPE`
 - `OADS_CONN_INFO_CLIENT_PROD_VERSION`
- Enhanced the `OASQLIP_set_connect_info` API to retrieve the cipher suite and cryptoprotocol version being used to connect to the OpenAccess SDK Server with the following info types:
 - `OADS_CONN_INFO_SESSION_CIPHER_SUITE`
 - `OADS_CONN_INFO_SESSION_CRYPTOPROTOCOL_VERSION`

Please update this API in the IP

Java SQL IP

Refer to the OpenAccess™ SDK Third-Party SQL Engine Programmer's Guide for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code to the new folder and make changes to the IP API as described below.

IP API Changes

- Enhanced the `sqlipSetInfo` API to retrieve the client type and client version with the following info types:
 - `OADS_CONN_INFO_CLIENT_TYPE`
 - `OADS_CONN_INFO_CLIENT_PROD_VERSION`
- Enhanced the `sqlipSetInfo` API to retrieve the cipher suite and cryptoprotocol version being used to connect to the OpenAccess SDK Server with the following info types:
 - `OADS_CONN_INFO_SESSION_CIPHER_SUITE`
 - `OADS_CONN_INFO_SESSION_CRYPTOPROTOCOL_VERSION`

Please update this API in the IP

Branding and Distribution

No changes.

File Name Changes

- The following files are changed from ivoa26* to ivoa27* in ODBC driver.

ivmgan27.dll, ivoa27m.dll, ivmghu27.dll, ivoad27.dll, ddextwin.exe, ivmgsp27.dll, ivoas27.dll, ivoa27.dll, ivoa27.ini, ivldap27.dll

- OpenSSL binary changed from OpenSSL800 to OpenSSL810
openssl810.dll/.so
- All the below server configuration files (admin folder) are changed from 800 to 810.
startOpenAccessSDK810_C, startOpenAccessSDK810_C_SQL,
startOpenAccessSDK810_Java, startOpenAccessSDK810_Java_SQL,
stopOpenAccessSDK810_C, stopOpenAccessSDK810_C_SQL,
stopOpenAccessSDK810_Java, stopOpenAccessSDK810_Java_SQL

Configuration Changes

The following new attributes are configured in oadm.ini configuration file. Refer to the *OpenAccess SDK Administrator's Guide* for details on these attributes.

- ServiceNetworkBufferSize
- ServiceSSLVersions
- ServiceSQLUseIndexedCache

OpenAccess SDK™ Version 8.0

Highlights

This section highlights what's new in OpenAccess SDK 8.0:

OpenAccess SDK SQL Engine and IP SDK

- Support for 64-bit row count
- Correlated query performance improvements
- Support for Public/Global IP address for Client Identification
- Support parameters in select list
- Improved Performance of IN subquery processing
- Flexible Numeric arithmetic – support configurable MIN_SCALE
- Enhanced support for LONG data types
 - Enhanced conversion of LONG data types
 - Enhanced scalar function support of LONG data types
 - Bulk protocol support for Numeric Precision up to 127
- Minimum .NET framework supported has been upgraded to 4.0 and support for older versions is dropped.

OpenAccess SDK Server

- Addressed Security vulnerability issues.

OpenAccess SDK Client for JDBC

- Support for CipherSuites connection attribute for SSL and fixes for uniform SSL behaviour between ODBC and JDBC

OpenAccess SDK Client for ODBC

- Support for 64 bit row count

OpenAccess SDK Client for ADO

- Support for 64 bit row count

OpenAccess SDK Client for ADO.NET

- Support for 64 bit row count

Upgrading from OpenAccess™ SDK 7.2 to 8.0

This section provides details for upgrading from the OpenAccess SDK 7.2 release to the 8.0 release. If you are upgrading from an earlier version, first refer to the “[Upgrading from OpenAccess™ SDK 7.0.1 to 7.2](#)” section of this document and then return here.

OpenAccess SDK 8.0 Clients and 8.0 Servers are not backward compatible with earlier versions. The following sections detail the changes for each of the different parts of the product.

Documentation

New documentation is being released for OpenAccess SDK 8.0. All changes to the OpenAccess SDK Version 7.2 are described in the fixes_<component>.txt, or this document.

License Keys

OpenAccess SDK Version 8.0 requires new keys that are different from the keys for OpenAccess SDK 7.2 and earlier. Please contact Progress customer support to obtain new keys.

C/C++ IP for OpenAccess™ SDK SQL Engine

Refer to the OpenAccess SDK SQL Engine Programmer's Reference for C/C++ for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code to the new folder and make changes to the IP APIs and DAM/DAMEX APIs as described in the following sections.

IP API Changes

1. Following Interface Provider APIs signatures are modified to support the 64-bit row count. The NumResRows output argument is a int64 value.

- *OAIP_execute(..,int64 *piNumResRows);*
- *OAIP_procedure(.., int64 *piNumResRows);*
- *OAIP_procedure_dynamic(..,int64 *piNumResRows);*
- *OAIP_ddl(.., int64 *piNumResRows);*
- *OAIP_dcl(.., int64 *piNumResRows);*
- *OAIP_native(.., int64 *piNumResRows);*
- *OAIP_nativeW(.., int64 *piNumResRows);*
- *OAIP_execute_dataTypes(.., int64 *piNumResRows);*

Please update these APIs in the IP.

2. Following is the new Information Type for OAIP_getInfo introduced in 8.0
 - **IP_INFO_MINIMUM_NUMERIC_SCALE**
Return the min scale value for NUMERIC/DECIMAL data types to OpenAccess SDK SQL engine to perform numeric calculations. If this value is not specified in IP, a default value of 3 is returned. An error is reported if the value is greater than 127 or is less than 0.

DAM API changes

1. Following is the list of DAM APIs, which are modified to support the 64-bit cardinality values.

- *dam_add_damobj_stat(.., int64 cardinality,..)*
- *dam_add_damobj_statW(.., int64 cardinality, ..)*

2. Information Types for `dam_getInfo` -

In 8.0, `dam_getInfo` () will return a 64 bit integer value for the information types related to row count. IP code needs to be modified to pass a pointer to a 64 bit integer to `dam_getInfo()` to retrieve the values for the below information types.

- *DAM_INFO_QUERY_TOP_ROWS*
- *DAM_INFO_QUERY_MAX_ROWS*
- *DAM_INFO_OUTER_TABLE_CUR_ROWCOUNT*

Apart from this, below are the new information types introduced in 8.0

- *DAM_INFO_CLIENT_PUBLIC_ADDRESS*

3. DAMEX (passthrough) APIs

Below DAMEX API is modified to take pointer to `int64` variable for number of rows specified in the TOP clause,

- *damex_describeSelectTopClause* (*.., int64 *piNumRows...*);

Java IP for OpenAccess™ SDK SQL Engine

Refer to the OpenAccess SDK SQL Engine Programmer's Reference for Java for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code to the new folder and make changes to the IP APIs and DAM/DAMEX APIs as described in the following sections.

IP API Changes

1. Following is the list of Interface Provider APIs whose signatures are modified to support the 64-bit row count.

- *ipExecute*(*..., xo_long piNumResRows*)
- *ipProcedure*(*..., xo_long piNumResRows*)
- *ipProcedureDynamic*(*..., xo_long piNumResRows*)
- *ipDCL*(*..., xo_long piNumResRows*)
- *ipDDL*(*..., xo_long piNumResRows*)
- *ipNative*(*..., xo_long piNumResRows*)

Please update these APIs in the IP.

2. Following is the new Information Type for `ipGetInfo` introduced in 8.0

- *IP_INFO_MINIMUM_NUMERIC_SCALE*
Return the min scale value for NUMERIC/DECIMAL data types to OpenAccess SDK SQL engine to perform numeric calculations. If this value is not specified in IP, a default value of 3 is returned. An error is reported if the value is greater than 127 or is less than 0.

DAM API Changes

1. schemaobj_stat Class

This class is used to define table indexes and table statistics schema information. SetObjInfo method is used to set values for this object. Signature of SetObjInfo is modified as shown below to support larger(>2B) cardinality values.

- *SetObjInfo(..... , long lCardinality,....)*

2. Information Types for dam_getInfo

In 8.0, dam_getInfo() will return a long integer value for the information types related to row count. So, IP code needs to be modified to pass a xo_long object to dam_getInfo() to retrieve the values for the below information types.

- *DAM_INFO_QUERY_TOP_ROWS*
- *DAM_INFO_QUERY_MAX_ROWS*
- *DAM_INFO_OUTER_TABLE_CUR_ROWCOUNT*

Apart from this, below are the new information types introduced in 8.0

- *DAM_INFO_CLIENT_PUBLIC_ADDRESS*

3. DAMEX (passthrough) APIs

Below DAMEX API is modified to take xo_long object for number of rows specified in the TOP clause,

- *damex_describeSelectTopClause(....., xo_long piNumRows,....);*

.NET IP for OpenAccess™ SDK SQL Engine

Refer to the OpenAccess SDK SQL Engine Programmer's Reference for .NET for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code to the new folder and make changes to the IP APIs and DAM/DAMEX APIs as described in the following sections

IP API Changes

1. Following is the list of Interface Provider APIs whose signatures are modified to support the 64-bit row count.

- *ipExecute(...,out long piNumResRows)*
- *ipProcedure(..., out long piNumResRows)*
- *ipProcedureDynamic(..., out long piNumResRows)*
- *ipDCL(...,out long piNumResRows)*
- *ipDDL(...,out long piNumResRows)*
- *ipNative(..., out long piNumResRows)*

Please update these APIs in the IP.

2. Following is the new Information Type for ipGetInfo introduced in 8.0

- *IP_INFO_MINIMUM_NUMERIC_SCALE*
Return the min scale value for NUMERIC/DECIMAL data types to OpenAccess SDK SQL engine to perform numeric calculations. If this value is not specified in IP, a default value of 3 is returned. An error is reported if the value is greater than 127 or is less than 0.

Helper Class Methods

schemaobj_stat Class

This class is used to define table indexes and table statistics schema information. SetObjInfo method is used to set values for this object. Signature of SetObjInfo is modified as shown below to support larger(>2B) cardinality values.

- *public void SetObjInfo(..., long lCardinality, ...)*

Information Types for dam_getInfo

In 8.0, dam_getInfo() in .NET takes a new argument to return long integer values for the information types related to row count.

- *dam_getInfo(..., out long pInt64InfoValue);*

So, IP code needs to be modified to pass long integer reference to dam_getInfo() to retrieve the values for the below information types.

- *DAM_INFO_QUERY_TOP_ROWS*
- *DAM_INFO_QUERY_MAX_ROWS*
- *DAM_INFO_OUTER_TABLE_CUR_ROWCOUNT*

Apart from this, below are the new information types introduced in 8.0

- *DAM_INFO_CLIENT_PUBLIC_ADDRESS*

DAMEX (passthrough) APIs

Below DAMEX API is modified to take long variable for number of rows specified in the TOP clause,

- *damex_describeSelectTopClause(..., out long piNumRows, ...);*

C SQL IP

Refer to the OpenAccess™ SDK Third-Party SQL Engine Programmer's Guide for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code to the new folder and make changes to the IP APIs as described below.

IP API Changes

Following is the list of Interface Provider APIs whose signatures are modified to support the 64-bit row count.

- *OASQLIP_get_stmt_info(..., int64 *NumericValuePtr);*
- *OASQLIP_execute_direct_stmt(..., int64 *piNumResRows);*
- *OASQLIP_execute_stmt(..., int64 *piNumResRows);*

Please update these APIs in the IP.

Java SQL IP

Refer to the OpenAccess™ SDK Third-Party SQL Engine Programmer's Guide for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code to the new folder and make changes to the IP APIs as described below.

- *sqlipExecuteStmt(xo_long piNumResRows)*

Please update this API in the IP

Branding and Distribution

No changes.

File Name Changes

- The following files are changed from *ivoa25** to *ivoa26** in ODBC driver.

ivmgan26.dll, ivoa26m.dll, ivmgphu26.dll, ivoad26.dll, ddextwin.exe, ivmgsp26.dll, ivoas26.dll, ivoa26.dll, ivoa26.ini, ivldap26.dll

- OpenSSL binary changed from *OpenSSL700* to *OpenSSL800*
openssl800.dll/.so

- All the below server configuration files (admin folder) are changed from 720 to 800.

*startOpenAccessSDK800_C, startOpenAccessSDK800_C_SQL,
startOpenAccessSDK800_Java, startOpenAccessSDK800_Java_SQL,
stopOpenAccessSDK800_C, stopOpenAccessSDK800_C_SQL, stopOpenAccessSDK800_Java,
stopOpenAccessSDK800_Java_SQL*

- Removed the following binary in 8.0 as stopped supporting .NET framework 3.5 and below.
oadamipnet.dll

Configuration Changes

No changes required to the *oadm.ini* configuration file.

OpenAccess SDK™ Version 7.2.0

Highlights

OpenAccess SDK 7.2.0 provides the following features.

OpenAccess SDK SQL Engine and IP SDK

- Increased performance for Big Data sets.
 - Support for DataDirect Bulk Load.
 - Support for disk-based nested query processing.
- Enhanced SQL support and query processing.
 - Support for expressions in the values of an INSERT statement.
 - Support for table functions in a FROM clause.
 - Support for TOP and ORDER BY clauses in a sub query.
 - Support for the CASE expression in GROUP BY, HAVING, and ORDER BY clauses.
 - Improved support for views.

Note: View definitions created with the earlier versions of OpenAccess SDK are not compatible with OpenAccess SDK 7.2. You must create all view definitions after setting up the schema folder with template schema files.

- Extended data types
 - Support for NUMERIC data type with 127 digit precision.
 - Support for DECIMAL data type with 127 digit precision.
- Enhanced analytic function support.
 - Support for Standard Deviation and Variance statistical functions.

OpenAccess SDK Server

- Embedded OpenAccess SDK clients to work for Windows non-administrator users.
- Addressed security vulnerability issues.

OpenAccess SDK Client for JDBC

- Added the DataDirect Bulk Load feature.
- OpenAccess SDK supports all JDKs from version 1.5.0_06 or higher.

OpenAccess SDK Client for ODBC

- Added the DataDirect Bulk Load feature.

OpenAccess SDK Client for ADO

- Fixed the branding tool for statement exit DLL.
- Added separate program groups for 32-bit and 64-bit ADO providers.

OpenAccess SDK Client for ADO.NET

No changes.

Upgrading from OpenAccess™ SDK 7.0.1 to 7.2.0

This section provides details for upgrading from the OpenAccess SDK 7.0.1 release to the 7.2.0 release. If you are upgrading from an earlier version, first refer to the [“Upgrading from OpenAccess™ SDK 7.0 to 7.0.1”](#) section of this document and then return here.

The OpenAccess SDK 7.2.0 clients are compatible with OpenAccess SDK 6.0, 6.0 SP1, 6.0 SP2, 6.0 SP3, 7.0 and 7.0.1 Servers. The OpenAccess SDK 6.0, 6.0 SP1, 6.0 SP2, 6.0 SP3, 7.0 and 7.0.1 clients are compatible with OpenAccess SDK 7.2.0 Servers.

The following sections detail the changes for each of the different parts of the product.

Documentation

New documentation is being released for OpenAccess SDK 7.2.0. All changes to the OpenAccess SDK Version 7.0.1 are described in the fixes.txt, or this document.

License Keys

OpenAccess SDK Version 7.2.0 requires new keys. Please contact Progress customer support to obtain new keys.

C/C++ IP for OpenAccess™ SDK SQL Engine

Refer to the *OpenAccess SDK SQL Engine Programmer’s Reference for C/C++* for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code to the new folder and make changes as described in the following sections.

IP API Changes

- Update IP_SUPPORT to set default values for new options. Sample IP code uses IP_SUPPORT_ARRAY (xxx_support_array) to set default values. Add new entries to xxx_support_array after IP_SUPPORT_JOIN_ORDER_SELECTION.

```
const IP_SUPPORT_ARRAY mem_support_array =
    0, /* IP_SUPPORT_UNICODE_INFO */
    0, /* IP_SUPPORT_JOIN_ORDER_SELECTION */
    0, /* IP_SUPPORT_TABLE_FUNCTIONS */
    0, /* IP_SUPPORT_BULK_INSERT */
    0, /* Reserved for future use */
    0, /* Reserved for future use */
```

Java IP for OpenAccess™ SDK SQL Engine

Refer to the *OpenAccess SDK SQL Engine Programmer’s Reference for Java* for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code to the new folder and make changes as described in the following sections.

IP API Changes

- Update IP SUPPORT to set default values for new options. Add new entries to IP_SUPPORT_ARRAY (xxx_support_array) after IP_SUPPORT_JOIN_ORDER_SELECTION.

```
private final int [] ip_support_array =
    0, /* IP_SUPPORT_UNICODE_INFO */
    0, /* IP_SUPPORT_JOIN_ORDER_SELECTION */
    0, /* IP_SUPPORT_TABLE_FUNCTIONS */
    0, /* IP_SUPPORT_BULK_INSERT */
    0, /* Reserved for future use */
    0, /* Reserved for future use */
```

JDK Version

OpenAccess SDK supports all JDKs from version 1.5.0_06 or higher.

.Net IP for OpenAccess™ SDK SQL Engine

Refer to the *OpenAccess SDK SQL Engine Programmer's Reference for .Net* for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code to the new folder and make changes as described in the following sections.

IP API Changes

- Update IP SUPPORT to set default values for new options. Add new entries to IP_SUPPORT_ARRAY (xxx_support_array) after IP_SUPPORT_JOIN_ORDER_SELECTION.

```
private int[] ip_support_array =
    0, /* IP_SUPPORT_UNICODE_INFO */
    0, /* IP_SUPPORT_JOIN_ORDER_SELECTION */
    0, /* IP_SUPPORT_TABLE_FUNCTIONS */
    0, /* Reserved for future use */
    0, /* Reserved for future use */
```

Branding and Distribution

No changes.

File Name Changes

No changes.

Configuration Changes

No changes required to the oadm.ini configuration file.

OpenAccess™ SDK Version 7.0.1

Highlights

OpenAccess SDK 7.0.1 provides the following features.

OpenAccess™ SDK SQL Engine and IP SDK

- Improved scalability.
 - Disk cache support for UNION query results.
- UNION query processing order is changed to LTR (Left-to-Right Subquery) as per the SQL standard.
- Support for custom scalar functions in WHERE clause.
- Support for aggregation functions in expressions.
- Java SQL (JSQL) support for Cancel.
- Java SE 7 certification on platforms for which Java SE 7 is available.
- Improved exception handling and error reporting.
 - Robust handling of "out of memory" conditions.
 - Report ambiguous columns in JOIN queries.
 - More descriptive error message when required IP functions are not implemented.
 - Addressed memory leak issues.

OpenAccess™ SDK Server

- Addressed security vulnerability issues.

OpenAccess™ SDK Client for JDBC

- Enhanced JDBC driver to recognize DataSourceFixCharTrim.
- JDBC driver synchronization issue in encryption UID/PWD

OpenAccess™ SDK Client for ODBC

- Added support for NULL data type.

OpenAccess™ SDK Client for ADO

No changes.

OpenAccess™ SDK Client for ADO.NET

No changes.

Upgrading from OpenAccess™ SDK 7.0 to 7.0.1

This section provides details for upgrading from the OpenAccess SDK 7.0 release to the 7.0.1 release. If you are upgrading from an earlier version, first refer to the “[Upgrading from OpenAccess SDK 6.0 SP3 to 7.0](#)” section of this document and then return here.

The OpenAccess SDK 7.0.1 clients are compatible with OpenAccess SDK 6.0, 6.0 SP1, 6.0 SP2, 6.0 SP3, and 7.0 Servers. The OpenAccess SDK 6.0, 6.0 SP1, 6.0 SP2, 6.0 SP3, and 7.0 clients are compatible with OpenAccess SDK 7.0.1 Servers.

The following sections detail the changes for the different parts of the product.

Documentation

Completely new documentation is not being released for OpenAccess SDK 7.0.1. All changes to the OpenAccess SDK Version 7.0 documents are described in the oasupp_701.pdf, fixes_701.pdf, or this document.

License Keys

OpenAccess SDK Version 7.0.1 uses the same keys as OpenAccess SDK Version 7.0.

Branding and Distribution

No changes.

File Name Changes

No file name changes.

Configuration Changes

No changes required to the oadm.ini configuration file.

OpenAccess™ SDK Version 7.0

Highlights

OpenAccess SDK 7.0.0 provides the following new features:

OpenAccess™ SDK SQL Engine and IP SDK

- Improved performance and scalability.
- Java IP Block Transfer of Rows
- Streaming LOB Data from IP
- Cache Large Binary Data
- Cost Based Join Ordering
- IP defined JOIN Ordering
- Reverse Primary Key and Foreign Key Join Ordering
- Customization of driver and data type information for the Java and .NET IPs
- Searchable LOB Data
- SQL RANK support
- Reduced memory usage for building results. OpenAccess SDK SQL Engine uses 15-20% less memory for query results.

- Improved exception handling and error reporting. Robust handling of "out of memory" conditions
- .NET IP support on 64-bit platforms
- .NET IP support for .NET Framework Versions 2.0, 3.0, 3.5, 3.5 SP1, and 4.0

OpenAccess™ SDK Server

- SSL Server certificate support

OpenAccess™ SDK Client for JDBC

- SSL encryption
- SSL Server certificate authentication

OpenAccess™ SDK Client for ODBC

- SSL encryption
- SSL Server certificate authentication
- ODBCISQL support for encrypted user name and password
- Protection of sensitive information in memory

OpenAccess™ SDK Client for ADO

- Support on 64-bit platforms
- SSL encryption
- SSL Server certificate authentication
- New configuration manager
- New branding tool
- Customization of logon and setup dialogs
- Localization support

OpenAccess™ SDK Client for ADO.NET

- Provider support on 64-bit platforms
- Support for .NET Framework 2.0, 3.0, 3.5, 3.5 SP1, and 4.0
- Support for Visual Studio 2008 or higher
- SSL encryption
- SSL Server certificate authentication

Upgrading from OpenAccess™ SDK 6.0 SP3 to 7.0

This section provides details for upgrading from the OpenAccess SDK 6.0 SP3 release to the 7.0 release. If you are upgrading from an earlier version, first refer to the “[Upgrading from OpenAccess SDK 6.0 SP1 to 6.0 SP3](#)” or [Upgrading from OpenAccess SDK 6.0 SP2 to 6.0 SP3](#)” sections of this document and then return here.

The OpenAccess SDK 7.0 clients are compatible with OpenAccess SDK 6.0, 6.0 SP1, 6.0 SP2, and 6.0 SP3 Servers. The OpenAccess SDK 6.0, 6.0 SP1, 6.0 SP2 and 6.0 SP3 clients are compatible with OpenAccess SDK 7.0 Servers.

SSL Compatibility

When an OpenAccess SDK 6.0 client connects with Encrypted=1 to an OpenAccess SDK 7.0 server, the client will not read the new DataSourceSSLDataEncryption attribute and the whole session will be SSL encrypted. This matches with the OpenAccess SDK 6.0 behavior.

An OpenAccess SDK 7.0 server configured with SSL certificates means that non-anonymous cipher suites will be configured. The server reports this to the client. The OpenAccess SDK 6.0 client only supported the anonymous cipher suites; this means that no match in cipher suites is found, resulting in a connection that cannot be set up. If the OpenAccess SDK 7.0 server supports one anonymous cipher suite, this suite will be used for SSL encryption (without certificates).

License Keys

OpenAccess SDK Version 7.0 requires new keys that are different from the keys for OpenAccess SDK 6.0 SP1 and later. Please contact Progress customer support to obtain new keys.

New universal platform keys allow a single key to be used instead of platform-specific keys. If you are licensed for all GA platforms, please contact Progress customer support to obtain these new keys.

C/C++ IP for OpenAccess™ SDK SQL Engine

Refer to the *OpenAccess SDK SQL Engine Programmer’s Reference for C/C++* for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code to the new folder and make changes as described in the following sections.

IP API Changes

- Update IP_SUPPORT_ARRAY (xxx_support_array) to set default values for new options.

Add new entries to xxx_support_array after IP_SUPPORT_UNICODE_INFO.

```
const IP_SUPPORT_ARRAY mem_support_array=  
0, /* IP_SUPPORT_XA */  
0, /* IP_SUPPORT_QUERY_MODE_SELECTION */  
0, /* IP_SUPPORT_VALIDATE_SCHEMAOBJECTS_IN_USE */  
0, /* IP_SUPPORT_UNICODE_INFO */  
0, /* IP_SUPPORT_JOIN_ORDER_SELECTION */  
0, /* Reserved for future use */  
0, /* Reserved for future use */
```

- Modify the IP to expose the new interfaces for data type and driver information. In `xxx_drv.c`, remove the following line which included definitions for functions that returned data type and driver information.

```
#include "damipex.h"
```

Also add `xxx_info.c` to the project. Template `xxx_info.c` is available under `<install_dir>\ip\oac\template_c\src`

- Modify the API `OAIP_getInfo()` to indicate the support values for the following newly added options.

`IP_INFO_DDL_RESULT_ROWS.`

`IP_INFO_GENERATE_COL_NAME_FOR_EXP.`

- Modify code that uses deprecated functions.

`dam_describeCond`: Modify the IP to replace the usage of deprecated API

`dam_describeCond` with `dam_describeCondEx`.

C/C++ IP for Third-party SQL Engine

Modify deprecated constants for return codes.

- Use `OADS_XXXX` constants instead of `SQLDRV_XXXX` constants for return codes.

Use the following new constants:

- Use `OADS_SUCCESS` for `SQLDRV_SUCCESS`
- Use `OADS_SUCCESS WITH_INFO` for `SQLDRV_SUCCESS_WITH_INFO`
- Use `OADS_NO_DATA` for `SQLDRV_EOS`
- Use `OADS_BUFFER_FULL` for `SQLDRV_BUFFER_FULL`
- Use `OADS_ERROR` for `SQLDRV_ERROR`
- Use `OADS_NOT_AVAILABLE` for `SQLDRV_NOT_AVAILABLE`

Java IP for OpenAccess™ SDK SQL Engine

Refer to the *OpenAccess SDK SQL Engine Programmer's Reference for Java* for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code to the new folder and make changes as described in the following sections.

IP API Changes

- Update `IP_SUPPORT_ARRAY` (`xxx_support_array`) to set default values for new options.

Add new entries to `xxx_support_array` after `IP_SUPPORT_UNICODE_INFO`.

```
private final int [] ip_support_array =
0, /* IP_SUPPORT_XA */
0, /* IP_SUPPORT_QUERY_MODE_SELECTION */
0, /* IP_SUPPORT_VALIDATE_SCHEMAOBJECTS_IN_USE */
0, /* IP_SUPPORT_UNICODE_INFO */
0, /* IP_SUPPORT_JOIN_ORDER_SELECTION */
0, /* Reserved for future use */
0, /* Reserved for future use */
```

- The API OAIP_getInfo() supports the following newly added options.
IP_INFO_DDL_RESULT_ROWS
IP_INFO_GENERATE_COL_NAME_FOR_EXP
IP_INFO_TYPE_INFO
IP_INFO_DS_INFO

Java IP for Third-party SQL Engine

No changes.

.NET IP for OpenAccess™ SDK SQL Engine

Refer to the *OpenAccess SDK SQL Engine Programmer's Reference for .NET* for details on setting up the environment to code your IP. Note that the IP project can be setup to use .NET Framework 2.0, 3.0, 3.5, 3.5 SP1 or .NET Framework 4.0. Once you have set up the environment, copy your IP code to the new folder and make changes as described in the following sections.

IP API Changes

- .NET Service is now supported on both 32-bit and 64-bit platforms with the same interface definition. To enable this support, many IP methods that formerly took int type arguments for handles now take the int64 type arguments. All changed arguments are listed in blue.

Change from int to int64 for handles:

ipConnect

```
Public int ipConnect (Int64 tmHandle, Int64 dam_hdbc, String sDataSourceName,
String sUserName, String sPassword, , String sCurrentCatalog, String sIPProperties,
String sIPCustomProperties)
```

ipDisconnect

```
Public int ipDisconnect (Int64 dam_hdbc)
```

ipStartTransaction

```
Public int ipStartTransaction (Int64 dam_hdbc)
```

ipEndTransaction

```
Public int ipEndTransaction (Int64 dam_hdbc, int iType)
```

ipExecute

```
Public int ipExecute (Int64 dam_hstmt, int iStmtType, Int64 hSearchCol, out int
piNumResRows)
```

ipSchema

```
Public int ipSchema (Int64 dam_hdbc, Int64 pMemTree, int iType, Int64 pList, Object
pSearchObj)
```

ipDDL

```
Public int ipDDL (Int64 dam_hstmt, int iStmtType, out int piNumResRows)
```



```

ipProcedure
    Public int    ipProcedure (Int64 dam_hstmt, int iType, out int piNumResRows)

ipDCL
    Public int    ipDCL (Int64 dam_hstmt, int iStmtType, out int piNumResRows)

ipNative
    Public int    ipNative (Int64 dam_hstmt, int iCommandOption, String sCommand, out
        int piNumResRows)

ipSchemaEx
    Public int    ipSchemaEx (Int64 dam_hstmt, Int64 pMemTree, int iType, Int64 pList,
        Object pSearchObj)

```

- To enable support on 32-bit and 64-bit platforms, DAM methods of the format `dam_xxx()`, which formerly took `int` type arguments (for handles), now take the `int64` type arguments. Please refer to the *OpenAccess SDK SQL Engine Programmer's Reference for .NET* for the signatures of the `dam_xxx ()` methods.
- Update `IP_SUPPORT_ARRAY` (`xxx_support_array`) to set default values for new options.

Add new entries to `xxx_support_array` after `IP_SUPPORT_UNICODE_INFO`.

```

private int[] ip_support_array =
0, /* IP_SUPPORT_XA */
0, /* IP_SUPPORT_QUERY_MODE_SELECTION */
0, /* IP_SUPPORT_VALIDATE_SCHEMAOBJECTS_IN_USE */
0, /* IP_SUPPORT_UNICODE_INFO */
0, /* IP_SUPPORT_JOIN_ORDER_SELECTION */
0, /* Reserved for future use */
0, /* Reserved for future use */

```

- The API `OAIP_getInfo()` supports the following newly added options.
`IP_INFO_DDL_RESULT_ROWS`.
`IP_INFO_GENERATE_COL_NAME_FOR_EXP`
`IP_INFO_TYPE_INFO`
`IP_INFO_DS_INFO`

Branding and Distribution

All DLLs (libs) that are part of the Client *for* ODBC and Client *for* ADO are renamed when branding is performed. This allows customers to use drivers from multiple OpenAccess SDK OEM partners. The OEM prefix (2 characters) will prefix the dll driver names. For more information, refer to the *OpenAccess SDK Distribution Guide*.

File Name Changes

- OpenSSL 1.0.0 is used in OpenAccess SDK 7.0. The standard libraries are bundled and renamed (openssl700.dll/openssl700.so) to avoid name conflicts. The openssl700 dll or shared library is loaded dynamically. If SSL is not needed, you can omit this library from the install package.
- ICU has been upgraded to version 4.2.1 in OpenAccess SDK 7.0. The names of the ICU libraries in the install package have changed.

Configuration Changes

The following new attributes are configured in oadm.ini configuration file. Refer to the *OpenAccess SDK Administrator's Guide* for details on these attributes.

- ServiceTCPBindSpecific
- ServiceSSLKeyStore
- ServiceSSLKeyStorePasswordFile
- ServiceSQLBulkByteBufferSize
- ServiceSQLBulkStringBufferSize
- ServiceSQLResultBufferCacheMaxSize
- DataSourceMSSConnectOptions
- DataSourceSSLDataEncryption
- DataSourcePadBlanksUpToColumnSizeInBytes
- DataSourceDescribeWcharAsChar
- DataSourceDescribeVarcharAsLongVarchar

OpenAccess™ SDK Version 6.0 SP3

Highlights

This section highlights what's new in OpenAccess SDK 6.0 SP3:

- **IPv6 Support** – The ODBC, JDBC, and Server components support IPv6 and IPv4 addresses. IPv6 addresses are 128-bit identifiers for interfaces and sets of interfaces. See <http://tools.ietf.org/html/rfc4291#section-2.2> for more information.
- **Improved ODBC setup sample** – The ODBC setup sample has been enhanced to support all the functionality that is in the OpenAccess SDK version of the ODBC setup DLL.
- **Support for Microsoft Windows 7** – Support for Windows 7 has been added in this release. The binaries from OpenAccess SDK 6.0 SP1 and SP2 also work on Windows 7, but the installers for those service packs have issues.
- **Customization and Localization of error messages** – The error messages returned by the JDBC client, ODBC client, Server, and the SQL Engine can be localized or modified for one or more languages. This feature existed in OpenAccess SDK 5.6 but was not released in OpenAccess SDK 6.0 GA or SP1. For more information, refer to the Message Localization section in the *OpenAccess SDK Distribution Guide*.
- **Use of UTF-16 for Unicode data with the Third-Party SQL Engine API** – The Third-party SQL engine API for C allows Unicode data to be transferred as wchar_t or UTF-16. In earlier releases, only wchar_t encoding was supported.

Refer to the oareadme and fixes_60_sp3 files for more details on this release.

Upgrading from OpenAccess™ SDK 6.0 SP2 to 6.0 SP3

This section provides details for upgrading from the OpenAccess SDK 6.0 SP2 release to the 6.0 SP3 release. If you are upgrading from an earlier version, first refer to the Upgrading from OpenAccess SDK 6.0 SP1 to 6.0 SP3 section of this document and then return here.

The OpenAccess SDK 6.0 SP3 clients are compatible with OpenAccess SDK 6.0, 6.0 SP1, and 6.0 SP3 Servers. The OpenAccess SDK 6.0, 6.0 SP1, and 6.0 SP2 clients are compatible with OpenAccess SDK 6.0 SP3 Servers. The IP code developed for OpenAccess SDK 6.0 SP1 or SP2 works with OpenAccess SDK 6.0 SP3 without any modifications.

The OpenAccess SDK 6.0 SP3 Server and IP SDK for Windows and Client for ODBC cannot coexist with earlier 6.0 releases. Installing the 6.0 SP3 replaces an existing 6.0 installation. Please save your IP code and the configuration file oadm.ini to a separate location. Next, uninstall the earlier version, and then install OpenAccess SDK 6.0 SP3.

Follow the instructions in the *OpenAccess SDK Installation Guide* to install the OpenAccess SDK components. Refer to the *OpenAccess SDK Administrator's Guide* for details on configuring the OpenAccess SDK components.

The following sections detail the changes for each of the different parts of the product. Unchanged parts of the product are not listed.

Documentation

New documentation is not being released for OpenAccess SDK 6.0 SP3. All changes to the OpenAccess SDK Version 6.0 SP2 documents are described in the oareadme.txt, fixes_60_sp3.pdf, or this document.

License Keys

OpenAccess SDK Version 6.0 SP3 uses the same keys as OpenAccess SDK Version 6.0 SP1. OpenAccess SDK Version 6.0 SP1 introduced new keys that were different from the keys for OpenAccess SDK 6.0 GA for using the Local Client for ODBC and IP SDK and the Server and IP SDK on 64-bit platforms. For earlier releases, one key could be used for both 32-bit and 64-bit platforms. Please contact customer support to obtain new keys.

New universal platform keys allow a single key to be used instead of platform specific keys. If you are licensed for all GA platforms, please contact customer support to obtain these new keys.

C/C++ IP for OpenAccess™ SDK SQL Engine

No changes.

C/C++ IP for Third-party SQL Engine

No changes.

Java IP for OpenAccess™ SDK SQL Engine

No changes.

Java IP for Third-party SQL Engine

No changes.

.NET IP for OpenAccess™ SDK SQL Engine

No changes.

Branding and Distribution

Branding tools and the run-time installers have been improved. The tools and message files to support message localization are installed along with the SDK components. For more information, refer to the Message Localization section in the *OpenAccess SDK Distribution Guide*.

File Name Changes

No file name changes.

Configuration Changes

No changes required to the oadm.ini configuration file.

Upgrading from OpenAccess™ SDK 6.0 SP1 to 6.0 SP3

This section provides details for upgrading from the OpenAccess SDK 6.0 SP1 release to the 6.0 SP3 release. If you are upgrading from an earlier version, then first refer to the Upgrading to SP1 section of this document and then come back to here.

The OpenAccess SDK 6.0 SP3 clients are compatible with 6.0, 6.0 SP1, and 6.0 SP2 Servers. The OpenAccess SDK 6.0, 6.0 SP1, and 6.0 SP3 clients are compatible with 6.0 SP3 Servers. The IP code developed for OpenAccess SDK 6.0 SP1 works with 6.0 SP3 without any modifications.

The OpenAccess SDK 6.0 SP3 Server and IP SDK for Windows and Client for ODBC cannot coexist with earlier 6.0 releases. Installing the OpenAccess SDK 6.0 SP3 replaces an existing 6.0 installation. Please save your IP code and the configuration file oadm.ini to a separate location. Next, uninstall the earlier version and then install OpenAccess SDK 6.0 SP3.

Follow the instructions in the *OpenAccess SDK Installation Guide* to install the OpenAccess SDK components. Refer to the *OpenAccess SDK Administrator's Guide* for details on configuring the OpenAccess SDK components.

The following sections detail the changes for each of the different parts of the product.

Documentation

Updated documents were released in the OpenAccess SDK 6.0 SP2 release. New documentation is not being released for OpenAccess SDK 6.0 SP3. All changes to the SP2 documents are documented in the oareadme.txt, fixes_60_sp3.pdf, or this document.

License Keys

OpenAccess SDK Version 6.0 SP3 uses the same keys as 6.0 SP1. OpenAccess SDK Version 6.0 SP1 introduced newer keys than 6.0 GA for using the Local Client for ODBC and IP SDK and the OpenAccess SDK Server and IP SDK on 64-bit platforms. Earlier, one key could be used for both 32-bit and 64-bit platforms. Please contact Progress customer support to obtain new keys.

New universal platform keys allow a single key to be used instead of platform-specific keys. If you are licensed for all GA platforms, please contact customer support to obtain these new keys.

C/C++ IP for OpenAccess™ SDK SQL Engine

No changes.

C/C++ IP for Third-party SQL Engine

The definition of Unicode arguments in the Third-party SQL engine API have changed from OAWCHAR to OASQLIP_CHAR. OAWCHAR was defined as wchar_t *. OASQLIP_CHAR is defined as void * and is used to pass in Unicode arguments as wchar_t or UTF-16. The Third-party SQL engine IP must be modified for this change. If you want to continue using wchar_t as the encoding for Unicode data, then, in the body of your function, cast the passed in argument as wchar_t. If you want to use the UTF-16 encoding introduced in OpenAccess SDK 6.0 SP2, then cast the passed-in argument as short int *.

Java IP for OpenAccess™ SDK SQL Engine

No changes.

Java IP for Third-party SQL Engine

No changes.

.NET IP for OpenAccess™ SDK SQL Engine

No changes.

Branding and Distribution

The tools and message files to support message localization are installed along with the SDK components. For more information, refer to the Message Localization section in the *OpenAccess SDK Distribution Guide*.

File Name Changes

This section details the additions, name changes, and removal of run-time files in each of the components.

Component	Windows	Linux/UNIX
Local Client for ODBC /admin and /bin	libeasy32.dll → libddt32.dll	NA
Local Client for ODBC /admin and /bin	ssleay32.dll → ssl32.dll	NA
Local Client for ODBC /admin	oaclamsmsg.cat → mmcoamsmsg.dll	oaclamsmsg.cat → liboaclamsmsg.so
Local Client for ODBC /ip/bin	NA	libcudata.so.34
Local Client for ODBC /ip/bin	NA	libcuc.so.34
Local Client for ODBC /lib	NA	ivoa22.cat → libivoa22m.so ^{1,5}
Client for ODBC /lib	NA	ivoa22.cat → libivoa22m.so ^{1,5}
Client for ODBC /	libeasy32.dll → libddt32.dll	NA
Client for ODBC /	ssleay32.dll → ssl32.dll	NA
Server /admin and /bin	libeasy32.dll → libddt32.dll	NA
Server /admin and /bin	ssleay32.dll → ssl32.dll	NA
Server /admin	NA	liboaagentmsg.cat → liboaagentmsg.so ^{1,5}
Server /admin	NA	liboasoamsmsg.cat → liboasoamsmsg.so ^{1,5}
Server ip/bin	NA	libcudata.so.34
Server ip/bin	NA	libcuc.so.34

¹ .sl on HP-UX

² libvicu23.so only in lib in a 32-bit environment

³ libddicu22.so only in lib64 in a 64-bit environment

⁴ on AIX only

⁵ .a on AIX

Configuration Changes

The ServiceMessageFile service attribute, which applies to Local ODBC Client for ODBC and OpenAccess SDK Server, should be set to point to the new file names for messages. The oadm_ori.ini file has this entry setup correctly.

Component	Windows	Linux/UNIX
Local Client for ODBC	NA	libivoal22m.so ^{1,5}
Server	NA	liboasoamsg.so ^{1,5}

¹ .sl on HP-UX

² libivicu23.so only in lib in a 32-bit environment

³ libddicu22.so only in lib64 in a 64-bit environment

⁴ on AIX only

⁵ .a on AIX

OpenAccess™ SDK Version 6.0 SP1

Highlights

This section highlights what's new in OpenAccess SDK 6.0 SP1. If you are upgrading from 5.6 or earlier, read this section and then go to the OpenAccess SDK Version 6.0 Highlights section to learn about OpenAccess SDK 6.0 and how to upgrade to it from 5.6.

- **Local Client for ODBC and IP SDK on Linux and UNIX** – The local client for ODBC and IP SDK is now supported on Linux and UNIX platforms in both 32-bit and 64-bit. In earlier releases, the local solution was only supported on Windows.
- **Support for embedding the OpenAccess SDK Server** – The OpenAccess SDK Server can be embedded into a native or a Java process to allow the data processing to happen in the same address space as the data source.
- **Improved and extended branding support**– Branding tools have been extended and improved to allow branding the ODBC and JDBC clients on all platforms, branding the Management Console and Command Line, and branding Interactive SQL. Branding tools are now installed as part of the SDK installation.
- **ODBC client exits** – The ODBC client can be configured to call a custom `SQLDriverConnectW` and `SQLConnectW` function to allow additional processing to be done on the client side before processing the connection.
- **Customizable run-time installers** – Customizable run-time installers support the creation of run-time installers for ODBC client and OpenAccess Server on Windows. This feature allows installers to be created by editing a text file to change name of the product and other properties. The customizable run-time installers are installed as part of the SDK.
- **Universal platform license keys** – A single license key can enable the use of the ODBC client or the Server on every supported platform. In earlier releases, a unique key was required for each platform. Please contact customer support to obtain these new keys.
- **Cursor based processing of stored procedure results** – Results from a stored procedure can be returned in cursor mode. In earlier releases, the complete results from a stored procedure were collected in memory before sending them to the client.
- **Cursor based processing of aggregates** – Results from GROUP BY queries are sent to the client as soon as the number of groups specified by the fetch block size is collected in memory. In earlier releases, all groups were brought into memory before sending to the client and this limited the size of the GROUP BY results that could be handled.
- **Configuration parameters are centralized** – All configuration parameters are now stored in the OpenAccess SDK Configuration file `oadm.ini` and managed using the OpenAccess SDK Manager. In earlier releases, the configuration parameters for the OpenAccess SDK Server were stored in both the `oadmi.ini` file that is managed using the OpenAccess SDK Manager and in the `oasql.ini` file which had to be manually edited.

Upgrading from OpenAccess™ SDK 6.0 to 6.0 SP1

This section provides details for upgrading from the OpenAccess SDK Version 6.0 GA release to the 6.0 SP1 release. If you are upgrading from 5.6 or earlier, go to the section upgrading from OpenAccess SDK Version 5.6 to 6.0 and then refer to this section to make the changes to get to 6.0 SP1. Continue on to the instructions for [Upgrading from OpenAccess SDK 6.0 SP1 to SP3](#).

If you are implementing stored procedures, the IP code developed for OpenAccess SDK 6.0 GA requires modifications. Even if code change is not required, you must rebuild your IP for this new release. Installations for the OpenAccess SDK Server and the Local ODBC client must be modified for configuration changes.

Follow the instructions in the *OpenAccess SDK Installation Guide* to install the OpenAccess SDK components. Refer to the *OpenAccess SDK Administrator's Guide* for details on configuring the OpenAccess SDK components.

The following sections detail the changes for each of the different parts of the product. Parts that are not listed had no changes.

Documentation

The OpenAccess SDK documentation has not been modified for the SP1 release. All documentation changes are documented in the oareadme or this upgrade document.

License Keys

OpenAccess SDK Version 6.0 SP1 requires new 64-bit keys for using the Local Client for ODBC and IP SDK and the Server and IP SDK on a 64-bit platform. In earlier releases, one key could be used for both 32-bit and 64-bit platforms. Please contact customer support to obtain new keys.

New universal platform keys allow single key to be used instead of platform specific keys. If you are licensed for all GA platforms, then please contact customer support to obtain these new keys.

Configuration Changes

The OpenAccess SDK components no longer use the oasql.ini file for configuration parameters for the OpenAccess SDK SQL engine and for the interface to a third-party SQL engine. These parameters have been renamed and made part of the service attributes that are managed by the OpenAccess SDK Manager and stored in the oadm.ini file. The new service attributes are documented in the *OpenAccess SDK Administrator's Guide*.

If you are using non-default settings, then you must modify your installation scripts to add the service attributes using the OpenAccess SDK Manager instead of writing them into the oasql.ini file.

Table 1: Mapping from oasql.ini Entries to Service Attributes

Entry name in OASQL.INI (In 6.0 GA and earlier)	Service Attribute (OA 6.0 SP1)
LogFileClose	ServiceIPLogFileClose
LogFileMaxSize	ServiceIPLogFileMaxSize
CacheBlockSize	ServiceSQLDiskCacheBlockSize
CacheMemSize	ServiceSQLDiskCacheMemSize
CacheOptions	ServiceSQLDiskCachePath, ServiceSQLDiskCacheInitialSize, ServiceSQLDiskCacheIncrementSize, ServiceSQLDiskCacheMaxSize, ServiceSQLDiskCacheDataBlockSize
ConfigureSQLLimits	ServiceSQLConfigureSQLLimits
CustomErrorFile	ServiceSQLCustomErrorFile
FETCHBLOCK_SIZE	ServiceSQLFetchBlockSize
IPMessagePrefix	ServiceSQLIPMessagePrefix
JOINBLOCK_SIZE	ServiceSQLJoinBlockSize
JoinInQueryOrder	ServiceSQLJoinInQueryOrder
JoinOrderUsingSearchCondition	ServiceSQLJoinOrderUsingSearchCondition
PageSize	ServiceSQLXmPageSize
QueryMaxSize	ServiceSQLQueryMaxSize

ResultCacheRowBlockSize	ServiceSQLResultCacheRowBlockSize
ResultCacheValBlockSize	ServiceSQLResultCacheValBlockSize
SchemaCacheSize	ServiceSQLBinarySchemaCacheSize
SchemaUpdateAllowed	ServiceSQLSchemaUpdateAllowed
SetLocale	ServiceSQLSetLocale
SQLExceptionErrorPrefix	ServiceSQLExceptionErrorMessagePrefix
SQLExceptionVerboseMode	ServiceSQLExceptionVerboseMode
SubQueryTransform	ServiceSQLSubQueryTransform
SupportCheckConnectionAlive	ServiceSQLCheckConnectionAlive
SupportLocaleDecimalInQuery	ServiceSQLLocaleDecimalInQuery
SupportQueryCancel	ServiceSQLQueryCancel
TranslatedQueryMaxSize	ServiceSQLTranslatedQueryMaxSize
UsePages	ServiceSQLXmUsePages
ViewToSubQueryTransform	ServiceSQLViewToSubQueryTransform
HideQualifierOwner	ServiceSQLHideQualifierOwner
ByteBufferSize	ServiceJavaIPByteBufferSize
CatalogTablePrefix	ServiceJavaIPCatalogTablePrefix
DBMSNAME	ServiceJavaIPDBMSName
SigSegv	ServiceJavaIPCatchSigSegv
StringBufferSize	ServiceJavaIPStringBufferSize
UseBasicGetColVal	ServiceJavaIPUseBasicGetColVal
UseBulkFetch	ServiceJavaIPUseBulkFetch
UseCatalogQueries	ServiceJavaIPUseCatalogQueries

C/C++ IP for OpenAccess™ SDK SQL Engine

Summary of changes:

- IP API function OAIP_procedure has a new argument iType to allow the stored procedure to work in cursor mode. Refer to the README for details.
- IP API function OAIP_procedure_dynamic argument iType has additional values DAM_FETCH and DAM_CLOSE. Refer to the README for details.
- Added functions to the SQL Engine API to support BIGINT data type. See the README for details.

Refer to the *OpenAccess SDK SQL Engine Programmer's Reference for C/C++* for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code over to the new folder and make changes as described in the following sections.

IP API Changes

Arguments to OAIP_procedure have changed. See the README for details.

The iType argument to OAIP_procedure_dynamic takes additional values. See the README for details.

Function OAIP_getInfo supports newly added options IP_INFO_VALIDATE_SCALAR_FUNC, IP_INFO_IN_COND_LIST_NORMALIZATION_LIMIT, and IP_INFO_VALIDATE_QUERY_SEMANTICS. See the README for details.

Java IP for OpenAccess™ SDK SQL Engine

Summary of changes:

- IP API method ipProcedure has a new argument iType to allow the stored procedure to work in cursor mode. Refer to the README for details.
- IP API method ipProcedureDynamic argument iType has additional values DAM_FETCH and DAM_CLOSE. Refer to the README for details.
- Added functions to the SQL Engine API to support BIGINT data type. See the README for details.

Refer to the *OpenAccess SDK SQL Engine Programmer's Reference for Java* for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code over to the new folder and make changes as described in the following sections.

IP API Changes

Arguments to ipProcedure have changed. See the README for details.

The iType argument to ipProcedureDynamic takes additional values. See the README for details.

Method ipGetInfo supports newly added options IP_INFO_VALIDATE_SCALAR_FUNC, IP_INFO_IN_COND_LIST_NORMALIZATION_LIMIT, and IP_INFO_VALIDATE_QUERY_SEMANTICS. See the README for details.

.NET IP for OpenAccess™ SDK SQL Engine

Summary of changes:

- IP API method `ipProcedure` has a new argument `iType` to allow the stored procedure to work in cursor mode. Refer to the README for details.
- IP API method `ipProcedureDynamic` argument `iType` has additional values `DAM_FETCH` and `DAM_CLOSE`. Refer to the README for details.
- Added functions to the SQL Engine API to support `BIGINT` data type. See the README for details.

Refer to the *OpenAccess SDK SQL Engine Programmer's Reference for Java* for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code over to the new folder and make changes as described in the following sections.

IP API Changes

Arguments to `ipDynamic` have changed. See the README for details.

The `iType` argument to `ipProcedureDynamic` takes additional values. See the README for details.

Method `ipGetInfo` supports newly added options `IP_INFO_VALIDATE_SCALAR_FUNC`, `IP_INFO_IN_COND_LIST_NORMALIZATION_LIMIT`, and `IP_INFO_VALIDATE_QUERY_SEMANTICS`. See the README for details.

Branding and Distribution

Branding tools are now installed along with the SDK components. Refer to the *OpenAccess SDK Distribution Guide* for more details. In earlier releases, the branding tools had to be downloaded from the OEM ftp site.

Customizable run-time installers are installed along with the SDK components. Refer to the *OpenAccess SDK Distribution Guide* for more details.

OpenAccess SDK™ Version 6.0

Highlights

- **Completely new clients, server, and wire-level protocol** – OpenAccess™ SDK 6.0 leverages the industrial-strength technology of DataDirect Technologies® to provide a custom driver development kit with parts that contain technologies from the DataDirect Connect and SequeLink product families. These are the products that are used by millions of users to connect to Oracle, DB2, SQL Server, and other databases.
- **100% Java JDBC driver** – The JDBC client is 100% Java and JDBC 3.0 compliant.
- **100% managed code ADO.NET provider** - The ADO.NET provider is 100% managed code and ADO.NET 1.1 compliant.
- **High-level of compliance to standards** – The clients have been implemented for extensive coverage of the respective standard and verified for compliance by using DataDirect's extensive test suites.
- **Java on 64-bit platforms** – Writing an IP in Java is now supported on 64-bit HP-UX, AIX, Solaris, Linux, and Windows.
- **Management Console and Agent** – The OpenAccess SDK Server can now be managed from a remote Windows-based management console or a command line.
- **SSL** – OpenAccess SDK 6.0 Version 6.0 implements the SSL protocol for data encryptions and key exchange. Previous releases supported the use of AES-128 algorithm for data encryption.
- **Process per connection support on Windows** – Earlier releases supported process per connection (forking) only on UNIX. Now on Windows and UNIX, you can configure the use of process per connection, thread per connection, or a thread pool.
- **Large file support for disk cache** – Disk cache files can grow beyond 2 GB to allow processing larger queries.
- **Modular deployment** – The OpenAccess SDK SQL engine and the IP code are DLL/shared libraries. This modular architecture allows portions of a deployment to be upgraded without having to rebuild a single process that contains the SDK code and the custom IP code.

Upgrading from OpenAccess™ SDK 5.6 to 6.0

The OpenAccess SDK 6.0 Clients are not compatible with OpenAccess SDK 5.x Servers and OpenAccess SDK 5.6 Clients are not compatible with OpenAccess SDK 6.0 Servers. You must upgrade both the OpenAccess SDK Client and OpenAccess SDK Server to OpenAccess SDK 6.0.

Follow the instructions in the *OpenAccess SDK Installation Guide* to install the OpenAccess SDK Server and IP SDK and the client. Refer to the *OpenAccess SDK Administrator's Guide* for details on configuring the OpenAccess SDK Server to use your IP.

Although many parts of the OpenAccess SDK 6.0 are new, the API for the IP and the API for the OpenAccess SDK SQL engine (previously referred to as the DAM) for most part remain unchanged. The following sections detail the changes for each of the different configurations.

Configuration Changes

The OpenAccess SDK components no longer use the `openrda.ini` file for configuration parameters for the client and server components. All configuration parameters for the OpenAccess SDK SQL engine and for the interface to a third-party SQL engine have been renamed and made part of the Service attributes that are managed by the OpenAccess SDK Manager and stored in the `oadm.ini` file. The data source and server configuration information from `oadrd.ini` file has also been moved to the `oadm.ini` file. The new service attributes are documented in the *OpenAccess SDK Administrator's Guide*. For the mapping from the older names to the new Service attribute names, refer to the "[Configuration Changes](#)" section under the "Upgrading from 6.0 to 6.0 SP1".

C/C++ IP for OpenAccess™ SDK SQL Engine

Summary of changes:

- IP API function names have changed and some take additional arguments.
- SQL engine API related to schema objects has changes to support Unicode for all fields.
- The IP must be compiled into a DLL (shared library on UNIX).

Refer to the *OpenAccess SDK SQL Engine Programmer's Reference for C/C++* for details on setting up the environment to code your IP. Once you have set up the environment, copy your IP code over to the new folder and make changes as described in the following sections.

IP API Changes

The IP API now has pre-defined function names. In earlier releases, the IP used arbitrary function names and registered them with the OpenAccess SDK SQL engine using the `ipRegisterInterface` function. For example, the IP may have implemented a function called `myConnect` and registered as the IP CONNECT operation. The IP must now implement `OAIP_connect()` function and export it. Refer to the *OpenAccess SDK SQL Engine Programmer's Reference for C/C++* for details on for the IP API function names.

Your IP code must be changed to use the IP API function names detailed in Chapter 2 of the *OpenAccess SDK SQL Engine Programmer's Reference for C/C++*.

Remove `ipRegisterInterface` as it is no longer called.

The arguments to functions `OAIP_connect(W)`, `OAIP_dcl`, and `OAIP_ddl` have changed. `OAIP_connect` has additional arguments for the custom connection and IP properties. `OAIP_dcl` and `OAIP_ddl` can now return the number of rows affected.

The `OAIP_getInfo` and `OAIP_setInfo` work with short and int data types instead of `UInt16` and `UInt32` data types. Modify your `OAIP_getInfo` and `OAIP_setInfo` functions to cast the `pInfoValue` accordingly.

Update `IP_SUPPORT_ARRAY` (`xxx_support_array`) to set default values for new options. Add new entries to `xxx_support_array` after `IP_SUPPORT_BLOCK_JOIN`. Refer to the 6.0 samples for updated support array settings.

Arguments to OAIP_procedure have changed. See the README for details.

The iType argument to OAIP_procedure_dynamic takes additional values. See the README for details.

SQL Engine API Changes

In calls to dam_addError, use DAM_IP_ERROR for error index.

The functions used to add schema objects now use Unicode strings for fields such as TableType, Remarks. The following functions have been changed to have all Unicode strings:

- dam_add_damobj_tableW, dam_add_damobj_columnW, dam_add_damobj_procW
- dam_add_damobj_proc_columnW, dam_set_damobj_tableW
- dam_set_damobj_columnW, dam_set_damobj_procW, dam_set_damobj_proc_columnW

Table schema object does not use the table_struct field.

dam_set_damobj_table() does not have an argument for table_struct.

Changes to Info Types in the dam_getInfo:

- Use DAM_INFO_OASQL_INI instead of DAM_INFO_OPENRDA_INI
- DAM_INFO_CONNECT_STRING is not supported as Connect string is passed to the OAIP_connect(W)

Functions that return information about tables now use OAWCHAR for fields that were char *:

- dam_describeTableW - pTablePath argument is OAWCHAR *
- damex_describeTableW - pTablePath argument is OAWCHAR *
- dam_describeJoinTableW - pTablePath argument is OAWCHAR *
- dam_describeTableByProcessOrderW - pTablePath argument is OAWCHAR *

dam_describeColDetail – piSortOrder, piFuncType, piIndexType are removed.

The IP is compiled into a DLL/Shared Library

The IP code is now compiled into a DLL for Windows and a shared library for UNIX. The OpenAccess SQL Engine loads the IP module dynamically. In earlier releases, the IP code was linked with OpenAccess SDK libraries to create an executable (or a DLL for a local solution).

Your build files must be changed to build and place the IP module in *install_dir/ip/bin* folder. Start with the project/make files in the *install_dir/ip/oac/template_c* or *install_dir/ip/oac/template_cpp*. On certain platforms, including AIX and Windows, an export file must be modified to list all the IP API functions the IP is implementing. See the sample export file (.def file on Windows and .exp file on AIX) that is included with the SDK.

On Windows, the following preprocessor definitions must be used when building the IP sources:

```
WIN32, LINT_ARGS, USE_DLL, IMPORTDLL
```

Version 5.6 required only WIN32 and LINT_ARGS preprocessor definitions to be used.

Refer to the *OpenAccess SDK Administrator's Guide* for details on the configuration parameters that control the loading of the IP module.

C/C++ IP for Third-party SQL Engine (previously referred to as SQL IP)

Summary of changes:

- The IP API for implementing an interface to a third-party SQL engine has changed and requires the IP code to be modified.
- The IP must now be compiled into a DLL (shared library on UNIX).
- Data source information is now read from calls into `OASQLIP_get_ds_info`, instead of from `oainfo.ini` or `OA_INFO` table.
- Schema queries are mapped to calls into the IP API.
- IP API uses type defines in function declarations – for example, `OADS_INTEGER` instead of `int`.
- OpenAccess SDK Version 5.6 IP API functions `sqlip_set_info` and `sqlip_get_info` have been split into separate functions for data source, connection, and statement level options.
- The data types `BIT` and `TINYINT` have been added.

See *install_dir/ip/oasqlip/oaodbc* for an example that implements the modified IP API. It contains code to support both the use of metadata tables and ODBC API for retrieving schema information. Refer to the *OpenAccess SDK Third-party SQL Engine Programmer's Guide and Reference* for details on setting up to implement an IP in C or C++ and the IP API that must be implemented.

IP is compiled into a DLL/Shared Library

The IP code must be compiled into a DLL for Windows and a shared library for UNIX. The OpenAccess SDK Server loads the IP module dynamically. In earlier releases, the IP code was linked with OpenAccess SDK libraries to create an executable (or a DLL for a local solution).

Your build files must be changed to build and place the IP module in the *install_dir/ip/bin* folder. Start with the project/make files in the *install_dir/ip/oasqlip/oaodbc*. The *oaodbc* sample is an example of a third-party SQL Engine IP implemented using the ODBC API. On certain platforms, including AIX and Windows, an export file must be modified to list all the IP API functions the IP is implementing. See the sample export file (`.def` file on Windows and `.exp` file on AIX) that is included with the SDK.

On Windows, the following preprocessor definitions must be used when building the IP sources:

```
WIN32, LINT_ARGS, USE_DLL, IMPORTDLL
```

Version 5.6 required only `WIN32` and `LINT_ARGS` preprocessor definitions to be used.

IP API Changes

All the IP API function names now begin with `OASQLIP` instead of `sqlip`. For example, `sqlip_alloc_connect` is now `OASQLIP_alloc_connect`.

IP API arguments are defined in terms of typedefs like `OADS_INTEGER` and `OADS_SMALLINT`. Change your IP code to use these typedefs in the function declarations.

The following functions have been added (see oaodbc sample for their details):

- OASQLIP_get_ds_info – Use in place of queries to OA_INFO or reading from oainfo.ini.
- OASQLIP_get_connect_info – Retrieve connections level options.
- OASQLIP_set_connect_info – Set connection level options.
- OASQLIP_get_stmt_info – Retrieve statement level options.
- OASQLIP_set_stmt_info – Set statement level options.
- OASQLIP_cancel – Cancel a running query.
- OASQLIP_get_colattr – Retrieve column attributes that in OpenAccess SDK Version 5.6 were returned by get_colspec.
- OASQLIP_free_colval – This function is called after a call to OASQLIP_get_colval and should free up the buffer in which the data was returned.

The following functions have been renamed:

- sqlip_open_cursor has been renamed to OASQLIP_execute_stmt but its functionality remains the same.

The following functions have been deleted:

- sqlip_execute_stmt.
- sqlip_is_cursor_open.
- sqlip_declare_cursor.
- sqlip_set_info – has been split into OASQLIP_set_connect_info, OASQLIP_set_stmt_info.
- sqlip_get_info – has been split into OASQLIP_get_ds_info, OASQLIP_get_connect_info, and OASQLIP_get_stmt_info.

The following functions have changes in their arguments:

- OASQLIP_alloc_connect – First argument SQLS_HDBC replaced with OADS_HDBC.
- OASQLIP_connect – iAccessMode and pSqlSecData removed. Other arguments renamed to match corresponding configuration attribute names.
- OASQLIP_get_colspec – arguments piMantissaPrecision, piMaxExponent, pMemTree, pColAttributes, and piFags have been removed. Column attributes are now returned by the newly added OASQLIP_get_colattr function.

The following functions must be modified to handle added data types BIT and TINYINT:

- OASQLIP_init_params – add cases for XO_TYPE_BIT and XO_TYPE_TINYINT.
- OASQLIP_get_colval – return values for column of type XO_TYPE_BIT and XO_TYPE_TINYINT as a char.

The following functions have been added:

- OASQLIP_set_connect_info
- OASQLIP_set_stmt_info
- OASQLIP_get_colattr
- OASQLIP_cancel

Schema Access

Client requests for metadata result in a call to a metadata IP API call. For example, a client request for a list of tables results in a call to IP API function `OASQLIP_tables`. This function must return a result set that matches the schema documented for `OA_TABLES` in the *OpenAccess SDK Third-Party SQL Engine Programmer's Guide and Reference*.

Version 5.6 and earlier releases of OpenAccess SDK clients turned calls like `SQLTables()` into queries on metadata table `OA_TABLES`. Code was supplied that would allow these types of queries to be parsed and then mapped into function calls into the backend.

If your IP used this approach, then you can reuse those functions directly. If you relied on metadata tables to supply the metadata information, then you must convert calls to metadata IP API function such as `OASQLIP_tables` into a query on a view or table `OA_TABLES`. You can do this by using the code supplied in the `odbschema.c` file that is installed as part of the `oaodbc` example in `install_dir/ip/oasqlip/oaodbc`.

If you are using metadata tables:

- Columns have been added to `OA_TYPES`, `OA_COLUMNS`, `OA_PROCEDURECOLUMNS`, and `OA_FKEYS` metadata tables. Refer to the *OpenAccess SDK Third-Party SQL Engine Programmer's Guide and Reference* for changes in `OA_COLUMNS`, `OA_PROCEDURECOLUMNS`, `OA_FKEYS`, and `OA_TYPES` tables. You must expose these additional columns and populate them with appropriate data. If your data source does not support Unicode column types, then you can map all `WVARCHAR` column type to `VARCHAR`.
- Use the code in `odbschema.c` to handle metadata IP API calls. Refer to the `oaodbc` sample for an implementation. The `oaodbc` sample maps metadata calls to queries on metadata tables that are implemented as defined in our documentation as well as ODBC 3.0 specifications. Your IP must support: `OASQLIP_catalogs`, `OASQLIP_schemas`, `OASQLIP_tabletypes`, `OASQLIP_tables`, `OASQLIP_columns`, `OASQLIP_statistics`, `OASQLIP_procedures`, `OASQLIP_procedurecolumns`, `OASQLIP_specialcolumns`, `OASQLIP_foreignkeys`, `OASQLIP_primarykeys`, and `OASQLIP_types`.

Java IP for OpenAccess™ SDK SQL Engine

Summary of changes:

- Argument types and number to some of the methods in the `oajava.sql.ip` interface have changed.
- Argument types to some of the SQL engine (JDAM) methods have changed.
- Package name is changed from `oajava.ip` to `oajava.{example}`.

IP API Changes

Many methods that formerly took `int` type arguments now take the `long` type argument. This change was made to support 64-bit. All changed arguments are listed in blue.

Additional arguments:

- `ipConnect` – additional arguments
`public int ipConnect(long tmHandle, long dam_hdbc, String sDataSourceName, String`

sUserName, String sPassword, String sCurrentCatalog, String sIPProperties, String sIPCProperties)

- ipDDL
public int ipDDL(long dam_hstmt, int iStmtType, xo_int piNumResRows)
- ipDCL
public int ipDCL(long dam_hstmt, int iStmtType, xo_int piNumResRows)
- ipProcedure
public int ipProcedure(long dam_hstmt, int iType, xo_int piNumResRows)

Change from int to long for handles:

- ipDisconnect
public int ipDisconnect(long dam_hdbc)
- ipStartTransaction
public int ipStartTransaction(long dam_hdbc)
- ipEndTransaction
public int ipEndTransaction(long dam_hdbc, int iType)
- ipExecute
public int ipExecute(long dam_hstmt, int iStmtType, long hSearchCol, xo_int piNumResRows)
- ipSchema
public int ipSchema(long dam_hdbc, long pMemTree, int iType, long pList, Object pSearchObj)
- ipProcedure
public int ipProcedure(long dam_hstmt, int iType, xo_int piNumResRows)
- ipNative
public int ipNative(long dam_hstmt, int iCommandOption, String sCommand, xo_int piNumResRows)
- ipProcedureDynamic
public int ipProcedureDynamic(long dam_hstmt, int iType, xo_int piNumResRows)
- Scalar functions have long arguments
public long ip_func_xxx(long hstmt, long pMemTree, long hValExpList)

Other changes:

Update `ip_support_array` to set default values for new options. Add new entries to `ip_support_array` after `IP_SUPPORT_BLOCK_JOIN`. Refer to 6.0 samples for updated support array.

```
private final int[] ip_support_array =
    0, /* IP_SUPPORT_XA */
    0, /* IP_SUPPORT_QUERY_MODE_SELECTION */
    0, /* IP_SUPPORT_VALIDATE_SCHEMAOBJECTS_IN_USE */
    1, /* IP_SUPPORT_UNICODE_INFO */
```

- `ipGetInfo` should return null to accept a default value instead of an empty string.
- When checking for valid handles, your code must check value to be non zero instead of checking for greater than zero (> 0)

```
if(hindex.getVal() != 0).
```

Changes to the SQL engine (JDAM) methods

Every SQL engine (JDAM) method that takes or returns handles to OpenAccess SQL engine objects has been modified to use long instead of int or `xo_long` instead of `xo_int`. Please compile your code against the latest `oasql.jar` and fix these issues.

Refer to the *OpenAccess SDK SQL Engine Programmer's Reference for Java* for details on setting up the environment to code your IP.

Changes to constants:

- Option names for `ipGetInfo` have been renamed:
`IP_INFO_QUALIFIER_NAME` renamed as `IP_INFO_QUALIFIER_NAMEW`
`IP_INFO_OWNER_NAME` renamed as `IP_INFO_OWNER_NAMEW`
- Arguments to `dam_addError` have changed. Use `DAM_IP_ERROR` for error index in `dam_addError`

Changes to Schema methods:

- `Schemaobj_table`: Table schema object no longer uses `TableStruct` field. The following methods no longer need the `sTableStruct` argument:

`constructor` has one less argument

`SetObjInfo` has one less argument

Change to tracing methods:

- `tm_trace` method is no longer supported. Use the `trace` method.

Java IP for Third-party SQL engine (previously referred to as SQL IP)

Summary of changes:

- The IP API for implementing an interface to a third-party SQL engine has changed and requires the IP code to be modified.
- Data source information is not longer read from oainfo.ini or OA_INFO table but from calls into sqlipGetDSInfo.
- Schema information can be supplied using schema methods or by populating OpenAccess specific schema tables in the underlying database.
- Added data types BIT and TINYINT.
- sqlipGetInfo and sqlipSetInfo have changed arguments.

A fully functioning sample OAJAVA is installed in *install_dir/ip/oasqlip/oajava*. The oajava sample is an example of a third-party SQL Engine IP implemented using the JDBC API.

For more information, refer to the *OpenAccess SDK Third-party SQL Engine Programmer's Guide and Reference*.

IP API Changes

The following functions have been added (see oajava sample for their details):

- sqlipGetDSInfo: used in place of queries to OA_INFO or reading from oainfo.ini for SQLGetInfo type of calls.
- sqlipGetColAtt: retrieves column attributes that in version 5.6 were returned by sqlipGetColspec.

The following functions have been renamed:

- sqlipOpenCursor has been renamed to sqlipExecuteStmt but its functionality remains the same.

The following functions have been deleted:

- sqlipIsCursorOpen

The following functions have changes in their arguments:

- sqlipGetColspec: arguments piMantissaPrecision, piMaxExponent, pMemTree, pColAttributes, and piFags have been removed. Column attributes are now returned by the newly added sqlipGetColattr method.
- sqlipConnect: data source name and custom IP properties are passed in as arguments.
- sqlipSetInfo: arguments have changed.

The following functions need to be modified to handle added data types BIT and TINYINT:

- `sqlipInitParam`: add cases for `XO_TYPE_BIT` and `XO_TYPE_TINYINT`
- `sqlipGetColVal`: return values for column of type `XO_TYPE_BIT` and `XO_TYPE_TINYINT` as short int.
- `sqlipFetchRowsInBuffer`: return values for column of type `XO_TYPE_BIT` and `XO_TYPE_TINYINT` as short int.

Schema Access

Client requests for metadata result in a call to a metadata IP API call. For example, a client request for a list of tables results in a call to IP API function `sqlipTables`. This function must return a result set that matches the schema documented for `OA_TABLES` in the *OpenAccess SDK Third-Party SQL Engine Programmer's Guide and Reference*.

Version 5.6 and earlier releases of the OpenAccess SDK clients changed calls such as `SQLTables()` into queries on metadata table `OA_TABLES`. Code was supplied that allowed these types of queries to be parsed and then mapped into function calls to the backend data source. If your IP used this approach, then you can reuse those functions directly. The service attribute `ServiceJavaIPUseCatalogQueries` controls whether functions such as `sqlipTables()` is called or whether a query `SELECT ... FROM oa_tables ...` is created.

If you are using metadata tables:

- Columns have been added to `OA_TYPES`, `OA_COLUMNS`, `OA_PROCEDURECOLUMNS`, and `OA_FKEYS` metadata tables. Refer to the Schema Objects section in the *OpenAccess SDK Third-Party SQL Engine Programmer's Guide and Reference* for changes in `OA_COLUMNS`, `OA_PROCEDURECOLUMNS`, `OA_FKEYS`, and `OA_TYPES` tables. You must expose these additional columns and populate them with appropriate data. If your data source does not support Unicode column types, then you can map all `WVARCHAR` column type to `VARCHAR`.

.NET IP for OpenAccess™ SDK SQL engine

Summary of changes:

- Argument types and number to some of the methods in the interface `oanet.sql.ip` have changed.
- The IP project needs to reference `oadamipnet.dll` and `oanet.sql.ip` files.
- The resulting IP module must be located in `install_dir\ip\bin`.

IP API Changes

Additional arguments:

`ipConnect` – additional arguments that include the data source name and custom IP connection properties

© 2016 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

These materials and all Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. The information in these materials is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear therein. The references in these materials to specific platforms supported are subject to change.

Business Making Progress, Corticon, DataDirect (and design), DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, Deliver More Than Expected, Icenium, Kendo UI, Making Software Work Together, NativeScript, OpenEdge, Powered by Progress, Progress, Progress Software Business Making Progress, Progress Software Developers Network, Rollbase, RulesCloud, RulesWorld, SequeLink, Sitefinity (and Design), SpeedScript, Stylus Studio, TeamPulse, Telerik, Telerik (and Design), Test Studio, and WebSpeed are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries. AccelEvent, AppsAlive, AppServer, BravePoint, BusinessEdge, DataDirect Spy, DataDirect SupportLink, Future Proof, High Performance Integration, OpenAccess, ProDataSet, Progress Arcade, Progress Profiles, Progress Results, Progress RFID, Progress Software, ProVision, PSE Pro, SectorAlliance, Sitefinity, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, WebClient, and Who Makes Progress are trademarks or service marks of Progress Software Corporation and/or its subsidiaries or affiliates in the U.S. and other countries. Java is a registered trademark of Oracle and/or its affiliates. Any other marks contained herein may be trademarks of their respective owners.

Please refer to the Release Notes applicable to the particular Progress product release for any third-party acknowledgements required to be provided in the documentation associated with the Progress product.

Third Party Acknowledgments:

One or more products in the Progress DataDirect OpenAccess v8.1 release includes third party components covered by licenses that require that the following documentation notices be provided. If changes in third party components occurred for the current release of the Product, the following Third Party Acknowledgments may contain notice updates to any earlier versions provided in documentation or README file.

Progress DataDirect OpenAccess v8.1 incorporates OpenLDAP v2.2.6 library. Such technology is subject to the following terms and conditions: The OpenLDAP Public License Version 2.8, 17 August 2003.

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions in source form must retain copyright statements and notices,
2. Redistributions in binary form must reproduce applicable copyright statements and notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution, and
3. Redistributions must contain a verbatim copy of this document.

The OpenLDAP Foundation may revise this license from time to time. Each revision is distinguished by a version number. You may use this Software under terms of this license revision or under the terms of any subsequent revision of the license.

THIS SOFTWARE IS PROVIDED BY THE OPENLDAP FOUNDATION AND ITS CONTRIBUTORS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OPENLDAP FOUNDATION, ITS CONTRIBUTORS, OR THE AUTHOR(S) OR OWNER(S) OF THE SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The names of the authors and copyright holders must not be used in advertising or otherwise to promote the sale, use or other dealing in this Software without specific, written prior permission. Title to copyright in this Software shall at all times remain with copyright holders. OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Copyright 1999-2003 The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved. Permission to copy and distribute verbatim copies of this document is granted.

Progress DataDirect OpenAccess v8.1 incorporates OpenSSL toolkit v1.0.2_x. Such technology is subject to the following terms and conditions: LICENSE ISSUES ===== The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org. OpenSSL License -----

Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)" 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org. 5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project. 6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)" THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com). - Original SSLeay License -----

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved. This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL. This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com). Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the rouines from the library being used are not cryptographic related :-). 4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)" THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

Progress DataDirect OpenAccess v8.1 incorporates ICU v4.2.1. Such technology is subject to the following terms and conditions: ICU License - ICU 1.8.1 and later ICU License - ICU 1.8.1 and later COPYRIGHT AND PERMISSION NOTICE. Copyright (c) 1995-2010 International Business Machines Corporation and others. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder. All trademarks and registered trademarks mentioned herein are the property of their respective owners.