

Connection Reference

Appeon InfoMaker®

2017

DOCUMENT ID: DC33820-01-1700-01

LAST REVISED: July 2017

Copyright © 2017 by Appeon Limited. All rights reserved.

This publication pertains to Appeon software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Appeon Limited.

Appeon and other Appeon products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Appeon Limited.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP and SAP affiliate company.

Java and all Java-based marks are trademarks or registered trademarks of Oracle and/or its affiliates in the U.S. and other countries.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Appeon Limited, 1/F, Shell Industrial Building, 12 Lee Chung Street, Chai Wan District, Hong Kong.

Contents

About This Book	i
------------------------------	----------

CHAPTER 1	Database Parameters.....	1
	Database parameters and supported database interfaces	1
	AppDriverName.....	7
	AppName	7
	Async.....	8
	BindSPInput	10
	Block (ODBC, OLE DB, Oracle, and SNC)	12
	Block (DirectConnect and Adaptive Server Enterprise)	14
	CacheAuthentication	14
	CallEscape	15
	CharSet	16
	Client_Locale.....	17
	CnnPool.....	18
	CommitOnDisconnect	19
	ConnectAs.....	20
	ConnectOption	21
	ConnectionString.....	26
	CSIncr	27
	CSMax.....	27
	CSMin.....	28
	CursorLib.....	28
	CursorLock.....	29
	CursorScroll.....	30
	CursorType.....	31
	CursorUpdate	32
	Database	33
	DataLink	34
	DataSource	35
	Date.....	35
	DateFormat	38
	DateTime.....	39
	DateTimeAllowed	41

DateTimeFormat	42
DBGetTime.....	44
Db_Locale	45
DBTextLimit.....	46
DecimalSeparator.....	46
DefaultProcOwner	48
DelimitIdentifier.....	48
DelimitIdentifierToDB	50
DisableBind	51
Driver.....	54
DS_Alias.....	55
DS_Copy	57
DS_DitBase.....	59
DS_Failover.....	61
DS_Password.....	63
DS_Principal.....	65
DS_Provider	66
DS_TimeLimit.....	68
Encrypt	69
EncryptionPass	70
EncryptPassword	71
FailoverPartner	72
FoDelay	73
FoDialog	74
FoRetryCount.....	75
FormatArgsAsExp	75
GenerateEqualsNull	77
HANotification.....	77
Hint.....	78
Host.....	79
HostReqOwner.....	80
IdentifierQuoteChar	81
Identity.....	82
ImpersonationLevel	84
INET_DBPATH.....	85
INET_PROTOCOL	85
INET_SERVICE	86
Init_Prompt.....	87
InsertBlock.....	88
IntegratedSecurity	90
KeepAlive	90
Language	91
LCID	92
Locale.....	93

Location.....	94
Log	95
LoginTimeout.....	95
LowerCaseIdent	96
MapDateToDateTime	97
MaskPassword	97
MaxConnect	98
MixedCase	99
Mode	100
MsgTerse	101
NCharBind	102
NCharLiteral	103
NLS_Charset.....	104
NumbersInternal.....	105
NumericFormat.....	106
ObjectMode	108
OJSyntax.....	109
OnlineIndex	111
PackageProcs	112
PacketSize (ODBC).....	113
PacketSize (ASE, DIR, SNC, SYC).....	113
PBCatalogOwner.....	114
PBMaxBlobSize.....	117
PBMaxTextSize.....	117
PBNewSPInvocation	118
PBTrimCharColumns	119
PBUseProcOwner	120
PersistEncrypted	122
PersistSensitive	123
PoolCreator	123
Pooling	124
PoolPwd	125
Properties	125
ProtectionLevel.....	126
Provider	127
ProviderString.....	128
PWDialo g	130
PWEncrypt	131
QualifyPublic	132
Release	132
Request.....	135
RPCRebind	136
RTConnBalancing	137
Scroll	137

Sec_Channel_Bind.....	138
Sec_Confidential.....	140
Sec_Cred_Timeout.....	141
Sec_Data_Integrity.....	143
Sec_Data_Origin.....	144
Sec_Delegation.....	146
Sec_Keytab_File.....	147
Sec_Mechanism.....	149
Sec_Mutual_Auth.....	151
Sec_Network_Auth.....	153
Sec_Replay_Detection.....	155
Sec_Seq_Detection.....	156
Sec_Server_Principal.....	158
Sec_Sess_Timeout.....	160
ServerName.....	161
ServiceComponents.....	162
SessionHomogeneous.....	162
ShowTempTables.....	163
ShowWarnings.....	164
SPCache.....	165
SQLCache.....	166
SQLQualifiers.....	167
StatementCache.....	168
StaticBind.....	168
StmtCache.....	170
StrByCharset.....	171
StripParmNames.....	174
SvrFailover.....	175
SystemOwner.....	176
SystemProcs.....	177
TableCriteria.....	178
ThreadSafe.....	179
Time.....	181
TimeFormat.....	183
Timeout.....	184
TimeStamp.....	184
TraceFile.....	185
TrimSpaces.....	186
TrustedConnection.....	187
TrustServerCertificate.....	188
TRS.....	189
UnicharBind.....	190
URL.....	191
UseProcSyntax.....	192

	UTF8	193
CHAPTER 2	Database Preferences	195
	Database preferences and supported database interfaces	195
	AutoCommit.....	197
	Connect DB at Startup	199
	Connect to Default Profile	200
	Keep Connection Open	200
	Lock.....	201
	Read Only	204
	Shared Database Profiles	205
	SQL Terminator Character	206
	Use Extended Attributes.....	207
	Index	209

About This Book

- Audience** This book is for anyone who uses Appeon® InfoMaker® to connect to a database. It assumes that you are familiar with the database you are using and have installed the server and client software required to access the data.
- How to use this book** This book describes the database parameters and preferences you use to connect to a database in InfoMaker.
- Related documents** For information about connecting to a database in the InfoMaker development environment, see *Connecting to Your Database*.
For a complete list of InfoMaker documentation, see the preface of *Getting Started*.
- Other sources of information** Use the Appeon Product Manuals Web site to learn more about your product. The Appeon Product Documentation Web site is accessible using a standard Web browser.
To access the Appeon Product Documentation Web site, go to [Product Manuals](https://www.appeon.com/developers/library/product-manuals-for-im) at <https://www.appeon.com/developers/library/product-manuals-for-im>.
- Conventions** The formatting conventions used in this manual are:

Formatting example	Indicates
<code>Retrieve and Update</code>	When used in descriptive text, this font indicates: <ul style="list-style-type: none"> • Command, function, and method names • Keywords such as <code>true</code>, <code>false</code>, and <code>null</code> • Datatypes such as <code>integer</code> and <code>char</code> • Database column names such as <code>emp_id</code> and <code>f_name</code> • User-defined objects such as <code>dw_emp</code> or <code>w_main</code>
<i>variable</i> or <i>file name</i>	When used in descriptive text and syntax descriptions, oblique font indicates: <ul style="list-style-type: none"> • Variables, such as <i>myCounter</i> • Parts of input text that must be substituted, such as <i>pblname.pbd</i> • File and path names

Formatting example	Indicates
File>Save	Menu names and menu items are displayed in plain text. The greater than symbol (>) shows you how to navigate menu selections. For example, File>Save indicates “select Save from the File menu.”
<code>dw_1.Update ()</code>	Monospace font indicates: <ul style="list-style-type: none"> • Information that you enter in a dialog box or on a command line • Sample script fragments • Sample output fragments

If you need help

All customers are entitled to standard technical support for reproducible software defects. You can open a standard support ticket at the Appeon support site: <https://www.appeon.com/standardsupport/> (login required).

If your organization has purchased a premium support contract for this product, then the designated authorized support contact(s) may seek assistance with your technical issue or question at the Appeon support site: <https://support.appeon.com> (login required).

About this chapter

This chapter describes the syntax and use of each database parameter that you can set in InfoMaker.

Database parameters and supported database interfaces

The following table lists each supported database interface and the database parameters you can use with that interface in InfoMaker.

The database parameters are described in alphabetical order following the table.

Database interface	DBParm parameters	
ASE SAP Adaptive Server® Enterprise <hr/> Release The Release database parameter must be set to 15 or higher to use the ASE interface. <hr/>	AppName Async Block (DirectConnect and Adaptive Server Enterprise) CharSet CommitOnDisconnect CursorType CursorUpdate DateTimeAllowed DBGetTime DBTextLimit DelimitIdentifier DisableBind DS_Alias DS_Copy DS_DitBase DS_Failover DS_Password DS_Principal DS_Provider DS_TimeLimit FoDialog FormatArgsAsExp Host KeepAlive Language Locale Log	MaxConnect OJSyntax PacketSize (ASE, DIR, SNC, SYC) PBCatalogOwner PWDiallog PWEncrypt Release Sec_Channel_Bind Sec_Confidential Sec_Cred_Timeout Sec_Data_Integrity Sec_Data_Origin Sec_Delegation Sec_Keytab_File Sec_Mechanism Sec_Mutual_Auth Sec_Network_Auth Sec_Replay_Detection Sec_Seq_Detection Sec_Server_Principal Sec_Sess_Timeout ShowTempTables StaticBind SvrFailover SystemProcs TableCriteria TrimSpaces UTF8

Database interface	DBParm parameters	
DIR SAP Sybase DirectConnect™	AppName Async Block (DirectConnect and Adaptive Server Enterprise) CharSet CommitOnDisconnect CursorUpdate DateTimeAllowed DBGetTime DBTextLimit DecimalSeparator DelimitIdentifier FormatArgsAsExp HostReqOwner KeepAlive	Locale LowerCaseIdent MaxConnect PacketSize (ASE, DIR, SNC, SYC) PBCatalogOwner Request ShowWarnings SQLQualifiers StaticBind SystemOwner TableCriteria TrimSpaces TRS UseProcSyntax UTF8
IN9 Informix	Async CommitOnDisconnect DateTimeAllowed DBGetTime DecimalSeparator DelimitIdentifier DisableBind	INET_DBPATH INET_PROTOCOL INET_SERVICE OJSyntax Scroll ThreadSafe TrimSpaces
I10 Informix	Async Client_Locale CommitOnDisconnect DateTimeAllowed DBGetTime Db_Locale DecimalSeparator DelimitIdentifier DisableBind EncryptionPass Hint	INET_DBPATH INET_PROTOCOL INET_SERVICE OJSyntax OnlineIndex PBCatalogOwner Scroll StmtCache StrByCharset ThreadSafe TrimSpaces

Database interface	DBParm parameters	
JDB JDBC	Async CommitOnDisconnect Date DateTime DBGetTime DelimitIdentifier DisableBind Driver FormatArgsAsExp IdentifierQuoteChar LoginTimeout MsgTerse	MapDateToDateTime NumericFormat OJSyntax PBCatalogOwner Properties StaticBind TableCriteria Time TraceFile TrimSpaces URL
ODBC	Async Block (ODBC, OLE DB, Oracle, and SNC) CallEscape CommitOnDisconnect ConnectOption ConnectString CursorLib CursorLock CursorScroll Date DateTime DBGetTime DecimalSeparator DefaultProcOwner DelimitIdentifier DelimitIdentifierToDB DisableBind FormatArgsAsExp	IdentifierQuoteChar InsertBlock LoginTimeout MsgTerse NumericFormat OJSyntax PacketSize (ODBC) PBCatalogOwner PBNewSPInvocation PBTrimCharColumns PBUseProcOwner RPCRebind SQLCache StaticBind StripParmNames TableCriteria Time TrimSpaces

Using DBParms with ODBC
 These DBParm parameters are supported by the InfoMaker ODBC interface only if *both* the ODBC driver you are using and the back-end DBMS support the feature

Database interface	DBParm parameters	
OLE DB	Block (ODBC, OLE DB, Oracle, and SNC) CacheAuthentication CommitOnDisconnect DataLink DataSource DateFormat DateTimeFormat DecimalSeparator DelimitIdentifier DelimitIdentifierToDB DisableBind EncryptPassword IdentifierQuoteChar ImpersonationLevel IntegratedSecurity LCID Location	MaskPassword Mode OJSyntax PBCatalogOwner PBMaxBlobSize PBTrimCharColumns PersistEncrypted PersistSensitive Init_Prompt ProtectionLevel Provider ServiceComponents SPCache StaticBind TimeFormat Timeout TimeStamp
O90 Oracle9i O10 Oracle 10g ORA Oracle 11g/12c	AppDriverName (ORA only) Async Block (ODBC, OLE DB, Oracle, and SNC) CnnPool (deprecated) CommitOnDisconnect ConnectAs CSIncr (ORA only) CSMax (ORA only) CSMin (ORA only) Date DateTime DBGetTime DecimalSeparator DelimitIdentifier DisableBind FoDelay FoDialog FoRetryCount FormatArgsAsExp HANotification (O10 and later) MixedCase NCharBind NCharLiteral (O10 and later)	NLS_Charset NumbersInternal ObjectMode OJSyntax PackageProcs PBCatalogOwner PoolCreator (ORA only) Pooling (ORA only) PoolPwd (ORA only) PWDialog QualifyPublic RTConnBalancing (ORA only) SPCache ServerName (ORA only) SessionHomogeneous (ORA only) StatementCache (O10 and later) StaticBind StrByCharset SvrFailover TableCriteria ThreadSafe Time TimeStamp TrimSpaces

Database interface	DBParm parameters	
SNC SQL Native Client for Microsoft SQL Server	AppName Block (ODBC, OLE DB, Oracle, and SNC) CommitOnDisconnect Database DataLink DateFormat DBTextLimit DecimalSeparator DelimitIdentifier DisableBind Encrypt FailoverPartner Host Identity	OJSyntax PacketSize (ASE, DIR, SNC, SYC) PBCatalogOwner PBMaxBlobSize PBMaxTextSize Provider ProviderString SPCache StaticBind TimeFormat Timeout TrimSpaces TrustedConnection TrustServerCertificate TimeStamp
SYC SAP Adaptive Server® Enterprise <hr/> Set Release The Release database parameter must be set to the version of your Open Client® software (11 or higher) to use DS_* and Sec_* parameters.	AppName Async Block (DirectConnect and Adaptive Server Enterprise) CharSet CommitOnDisconnect CursorType CursorUpdate DateTimeAllowed DBGetTime DBTextLimit DelimitIdentifier DisableBind DS_Alias DS_Copy DS_DitBase DS_Failover DS_Password DS_Principal DS_Provider DS_TimeLimit FoDialog FormatArgsAsExp Host KeepAlive Language Locale Log	MaxConnect OJSyntax PacketSize (ASE, DIR, SNC, SYC) PBCatalogOwner PWDialog PWEncrypt Release Sec_Channel_Bind Sec_Confidential Sec_Cred_Timeout Sec_Data_Integrity Sec_Data_Origin Sec_Delegation Sec_Keytab_File Sec_Mechanism Sec_Mutual_Auth Sec_Network_Auth Sec_Replay_Detection Sec_Seq_Detection Sec_Server_Principal Sec_Sess_Timeout ShowTempTables StaticBind SvrFailover SystemProcs TableCriteria TrimSpaces UTF8

AppDriverName

Description Allows you to set your own client driver name for the Oracle database connection.

Controls ORA Oracle 11g/12c

Syntax AppDriverName =value

Parameter	Description
value	Specifies the name of the driver to use for an Oracle database connection.

Default None.

Usage The maximum length of the name is 8 characters. You can display the client driver name with the V\$SESSION_CONNECT_INFO or GV\$SESSION_CONNECT_INFO dynamic performance view queries.

Examples To specify a client driver name, type the driver name in the Application Driver Name text box on the Connection page of the Database Profile Setup dialog box.

AppName

Description If the DBMS supports it, specifies the application name you want to use when connecting to the database in InfoMaker.

When to specify AppName

You must specify the AppName parameter *before* connecting to the database.

Controls ASE, SYC SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect
SNC SQL Native Client for Microsoft SQL Server

Syntax AppName='application_name'

Default For Adaptive Server and DirectConnect, InfoMaker sets the CS_APPNAME connection property to InfoMaker, as follows:

```
AppName= ' InfoMaker '
```

For SQL Server, there is no default value for AppName.

Usage *Adaptive Server databases* It is useful to specify a different AppName value for each of your Adaptive Server applications. If you are an administrator, you can query the MASTER.DBO.SYSPROCESSES table to determine which applications are running on the database server. The value specified for AppName displays in the program_name column of the MASTER.DBO.SYSPROCESSES table, making it easy to identify the applications.

Examples **Example** To set the application name to Test, type the following in the Application Name box on the Network or System page in the Database Profile Setup dialog box:

Test

See also Host (applies only to ASE and SYC SAP Adaptive Server Enterprise)

Async

Description Allows you to perform asynchronous operations on your database in InfoMaker. You can also switch to another Windows process while the retrieval takes place.

By default, InfoMaker operates synchronously.

Controls ASE, SYC SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect
I10 Informix
IN9 Informix
JDB JDBC
ODBC (if driver and back-end DBMS support this feature)
O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12c

Syntax Async=*value*

Parameter	Description
<i>value</i>	A value specifying synchronous or asynchronous operation. Values are: <ul style="list-style-type: none">• 0 (Default) Synchronous operation• 1 Asynchronous operation

Default Async=0

Usage

Enabling asynchronous operation in InfoMaker is useful when you are executing a complex **SQL** statement that takes several minutes to return results. If the Async parameter is set to 1, you can do either of the following while the **SQL** statement is executing:

- Work in another window
- Cancel the statement before it retrieves the first row of data

When to set Async If you are communicating with the database in code, you can reset the Async value at any time before or after the Transaction object has connected to the database.

How data is retrieved When you retrieve data in a form or report, the following steps occur in order:

- 1 The database server compiles and executes the **SQL** statement.
- 2 InfoMaker retrieves (fetches) the first row of data.
- 3 InfoMaker retrieves each subsequent row of data.

What happens before the first row is retrieved While the server is compiling and executing the **SQL** statement and before InfoMaker retrieves the first row of data, you must have done *both* of the following to enable asynchronous operation (allowing you to cancel the current operation before it retrieves the first row of data):

- Coded a RetrieveRow event for the form or report (the code can contain only a comment)
- Set the Async parameter to 1

What happens after the first row is retrieved After the first row of data is retrieved and between subsequent row fetches, you must have done only the following to enable asynchronous operation:

- Coded a RetrieveRow event for the form or report

After the first row is retrieved, InfoMaker operates asynchronously *without your having to set the Async parameter to 1*, so you can cancel the current operation anytime after it retrieves the first row of data. Therefore, the Async parameter has no effect in InfoMaker after the first row of data is retrieved.

Examples

Example 1 To enable asynchronous operation, select the Asynchronous check box on the Transaction page in the Database Profile Setup dialog box.

Example 2 You can set the Async and DBGetTime parameters in a single DBParm statement. DBGetTime specifies the number of seconds you want InfoMaker to wait for a response from the DBMS when you retrieve rows in a form or report. To enable asynchronous operation and set the DBGetTime parameter to 20 seconds, select the Asynchronous check box and type 20 in the Number Of Seconds To Wait box on the Transaction page in the Database Profile Setup dialog box.

See also

DBGetTime

BindSPInput

Description

Specifies that InfoMaker bind input parameters in dynamic SQL statements when executing a stored procedure.

Controls

ADO.NET
 O90 Oracle9i
 O10 Oracle 10g
 SNC SQL Native Client for Microsoft SQL Server

Syntax

BindSPInput=*value*

Parameter	Description
<i>value</i>	Specifies whether you want to bind input parameters in dynamic SQL statements when executing stored procedures. Values are: <ul style="list-style-type: none"> • 0 (Default) InfoMaker does not bind parameters in dynamic SQL statements when executing stored procedures. • 1 InfoMaker binds parameters in dynamic SQL statements when executing stored procedures.

Default

BindSPInput=0

Usage

For SNC, when BindSPInput is set to 0, you can use the same syntax to declare a stored procedure in a script as you can when using the PowerBuilder OLE DB interface. When BindSPInput is set to 1, the SNC interface supports SQL Server large value datatypes as procedure IN/OUT parameters or function return values.

The syntax for declaring a procedure with SNC is:

```

DECLARE logical_procedure_name PROCEDURE FOR
    [@rc=]procedure_name
    {@param1=value1 [OUTPUT], @param2=value2 [OUTPUT], ...}
    {USING transaction_object};
    
```

[@rc=] indicates that you want to get the procedure's return value.

Use the keyword **OUTPUT** or **OUT** to indicate an output parameter if you want to get the output parameter's value.

If BindSPInput=0, *value1, value2,...* can be either PowerBuilder script variables or literal values. If BindSPInput=1, *value1, value2,...* must be PowerBuilder script variables. If you specify literal values, the interface returns a runtime error.

When you declare a dynamic **SQL** statement with a procedure, enter a question mark (?) for each IN/OUT parameter in the statement. Value substitution is positional. For examples, see Dynamic **SQL** Format 3 and 4 in the online Help.

For Oracle, set BindSPInput to 1 to ensure that CLOB, NCLOB, and BLOB parameters work correctly as stored procedure parameters.

For ADO.NET:

- When BindSPInput is set to 1, parameter values must be PowerBuilder script variables, not literal values.
- The IBM.Data.Informix driver (used to access an ADO.NET compliant Informix database) does not support the BindSPInput dbparm.
- The ADO.NET Microsoft SQL Server interface does not support Text, NText, or Image parameters. Use VarChar(max) or VarBinary(max) instead.

Examples

Setting BindSPInput To specify that InfoMaker should bind parameters in dynamic **SQL** statements when executing a stored procedure:

- **Database profile** Select the Bind Procedure Parameters check box on the Transaction page in the Database Profile Setup dialog box.
- **Application** Type the following in code:

```
SQLCA.DbParameter="BindSPInput=1"
```

Using the ADO.NET SQL Server interface Consider the following two SQL statement fragments:

```
create procedure p_1 (@inparm1 TEXT) AS
```

The preceding statement does not work if BindSPInput is set to 1, because the SQL Server interface does not support Text.

```
create procedure p_2 (@inparm1 VARCHAR(MAX)) AS
```

The preceding statement can work with BindSPInput set to 1, because the SQL Server interface does support VARCHAR(MAX).

See also

[DisableBind](#)

Block (ODBC, OLE DB, Oracle, and SNC)

Description

For those interfaces that support it, Block specifies the cursor blocking factor when connecting to a database. The blocking factor determines the number of rows that a report can fetch from the database at one time.

Using the Block parameter can improve performance when accessing a database in InfoMaker.

Controls

ODBC (if driver and back-end DBMS support this feature)

OLE DB

O90 Oracle9i

O10 Oracle 10g

ORA Oracle 11g/12c

SNC SQL Native Client for Microsoft SQL Server

Syntax

Block=*blocking_factor*

Parameter	Description
<i>blocking_factor</i>	The number of rows you want the report to fetch from the database at one time. The blocking factor can be a number from 1 to 1000, inclusive. To turn off block fetching, set Block to 1.

Default

The default value for the Block parameter depends on the DBMS you are accessing, as summarized in the following table:

DBMS	Block default value
ODBC	For most reports, the Block default value is the following, <i>up to a maximum of 32K per column</i> : <code>Block=1000</code> If you specified that the report should retrieve only as many rows as needed from the database (Retrieve.AsNeeded property), the Block default value is the following, <i>up to a maximum of 32K per column</i> : <code>Block=100</code>
OLE DB	InfoMaker sets the blocking factor to 1
Oracle	InfoMaker sets the blocking factor dynamically if one row size is too large. By default the blocking factor is 300 rows.

Using the default blocking factor

You should not have to set a non-default value for Block. In most cases, the default blocking factor used by InfoMaker should meet your needs.

Usage

Requirements for ODBC data sources To use the Block database parameter with an ODBC data source, your ODBC driver must:

- Be ODBC Version 2.0 compliant or higher, *and*
- Support the `SQLExtendedFetch` API call

The `SQL Anywhere®` ODBC driver that comes with InfoMaker meets both of these requirements.

For information about whether your ODBC driver meets these requirements, see the documentation that comes with your driver.

Determining the Block value for ODBC data sources InfoMaker searches the following in this order to determine the Block value for ODBC data sources:

- 1 The section for your database profile in the registry or the value of the Transaction object property (in an application)
- 2 The section for your ODBC driver in the `PBODB170` initialization file

If InfoMaker does not find a Block value in these locations, it uses the default Block value for the DBMS you are accessing.

Turning off block fetching To turn off block fetching for an ODBC data source or Oracle database, set the Block parameter to 1.

OLE DB and Microsoft SQL Server When you use the OLE DB database interface with a Microsoft `SQL Server` database and retrieve data into a report or use an embedded `SQL` cursor in a `SELECT` statement, server-side cursors are used to support multiple command execution. If this has a negative impact on performance, try increasing the size of the Block database parameter to 500 or more, or adding the following line to the [Microsoft `SQL Server`] section in the `PBODB170.INI` file to turn off server-side cursors:

```
ServerCursor='NO'
```

Examples

To set the blocking factor for reports to 50 rows:

- **Database profile** Type 50 in the Retrieve Blocking Factor box on the Transaction page in the Database Profile Setup dialog box.

Block (DirectConnect and Adaptive Server Enterprise)

Description Specifies the internal blocking factor used by the SAP Sybase Client Library (CT-Lib) interface when declaring a cursor. The blocking factor determines the number of rows fetched from the database at one time when CT-Lib makes a physical request for data.

The Block DBParm parameter applies only to declared cursors and *not* to reports.

Controls ASE, SYC, SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect

Syntax Block=*blocking_factor*

Parameter	Description
<i>blocking_factor</i>	The number of rows fetched from the database at one time when CT-Lib makes a physical request for data (default=100 rows)

Default Block=100

Examples To set the blocking factor to 1000 rows, type the following in the Retrieve Blocking Factor box on the Transaction page in the Database Profile Setup dialog box:

1000

CacheAuthentication

Description Specifies whether the OLE DB data provider can cache sensitive authentication information, such as a password, in an internal cache.

When to specify CacheAuthentication

You must specify the CacheAuthentication parameter *before* connecting to the database.

Controls OLE DB

Syntax CacheAuthentication='value'

Parameter	Description
value	<p>Specifies whether the OLE DB data provider can cache authentication information. Values are:</p> <ul style="list-style-type: none"> • True Tells the OLE DB data provider to cache information. • False (Default) Tells the OLE DB data provider not to cache information.

Default CacheAuthentication='False'

Examples To tell the OLE DB data provider to cache authentication information, select the Cache Authentication check box on the Security page in the Database Profile Setup dialog box.

See also [DataLink](#)
[IntegratedSecurity](#)

CallEscape

Description Controls whether the ODBC interface uses call escape syntax for stored procedure calls (the default) or converts the calls to driver-specific native SQL syntax before sending the command to the ODBC driver.

Controls ODBC (if driver and back-end DBMS support this feature)

Syntax CallEscape='value'

Parameter	Description
value	<p>Controls whether the ODBC interface uses call escape syntax for stored procedure calls or converts the calls to driver-specific native SQL syntax. Values are:</p> <ul style="list-style-type: none"> • Yes (Default) The ODBC interface uses call escape syntax for stored procedure calls • No The ODBC interface converts stored procedure calls to driver-specific native SQL syntax before sending the command to the ODBC driver

Default CallEscape='Yes'

Usage *When to use* Set CallEscape to No if the ODBC driver you are using expects to receive stored procedure calls in native (driver-specific) SQL syntax instead of in call escape syntax.

For information about the stored procedure call syntax your ODBC driver expects, see your vendor's driver documentation.

Level 2 or higher ODBC driver required To use the CallEscape parameter, your ODBC driver *must* meet Level 2 or higher API conformance requirements. CallEscape has no effect when you are using an ODBC driver that meets Core or Level 1 API conformance requirements.

Example of stored procedure call escape syntax The following example shows a call to a stored procedure named sp_test that uses call escape syntax:

```
{call sp_test(1,1)}
```

Examples

To convert stored procedure calls to native SQL syntax before sending the command to your ODBC driver, clear the Use Call Escape Syntax check box on the Syntax page in the Database Profile Setup dialog box.

CharSet

Description

Specifies the character set you want the SAP Sybase Open Client™ software to use when connecting to an SAP Adaptive Server Enterprise database or a database accessed through DirectConnect.

When to specify CharSet

You must specify the CharSet parameter *before* connecting to a database.

Controls

ASE, SYC SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect

Syntax

CharSet=*character_set*

Default

None

Usage

When you specify a value for CharSet, InfoMaker:

- Allocates a CS_LOCALE structure for this connection
- Sets the CS_SYB_CHARSET value to the character set you specify
- Sets the SQL Server CS_LOC_PROP connection property with the new locale information

Overriding the Locale parameter If you have previously set a value for the Locale parameter that includes settings for the language and character set you want to use, you can override the character set value by specifying a new value for the CharSet parameter and reconnecting to the database.

Unicode data access InfoMaker can access Unicode data in an Adaptive Server Enterprise (ASE) 12.5 or later Unicode database or in Unicode columns in ASE 12.5 or later. InfoMaker converts between double-byte character set (DBCS) data and Unicode automatically, provided that the CharSet and Language parameters are set with DBCS values (or the Locale parameter is set with DBCS values). For example:

```
CharSet='big5'
Language='tchinese'
```

Examples

To set the character set to iso_1, type *iso_1* in the Character Set box on the Connection page or Regional Settings page in the Database Profile Setup dialog box.

See also

Language
Locale

Client_Locale

Description

Client_Locale identifies the locale that the client application uses.

When to specify Client_Locale

You must specify the Client_Locale parameter *before* connecting to a database.

Controls

I10 Informix

Syntax

Client_locale=*language_territory.codeset*'

Parameter	Description
<i>language</i>	Two character name that represents the language for a specific locale, for example “en” for English.
<i>territory</i>	Two character name that represents the cultural conventions for a specific territory, for example “AU” for Australia.
<i>codeset</i>	Name of the code set that the locale supports, for example “utf8.”

Default

Based on the operating system’s locale.

Usage

The I10 native interface uses the Informix GLS (Global Language Support) API for global language support. Client_Locale specifies the value of the Informix environment variable CLIENT_LOCALE. The I10 interface uses this setting to access string data in an Informix database and to process SQL statements. If you do not set the DbParameter, the default client locale value is based on the OS locale.

For example, to access a database that has a Japanese SJIS locale, Client_Locale should be set to `ja_jp.sjis` on the client system.

For more information about the Informix CLIENT_LOCALE and DB_LOCALE environment variables, see the *IBM Informix GLS User's Guide*, currently available at the Informix library web site.

Examples

To set the character set to `en_us.utf8`, type `en_us.utf8` in the Client Locale box on the Regional Settings page in the Database Profile Setup dialog box.

See also

`Db_Locale`
`StrByCharset`

CnnPool

Description

Specifies whether Oracle should maintain connections in a pool. An Oracle connection pool is a group of reusable physical connections spanning several sessions and managed by the Oracle Call Interface (OCI). The CnnPool DBParm is maintained in the ORA driver for backward compatibility only. It is ignored if you use the Pooling DBParm.

By default, connection pooling is not used.

Controls

O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12c

Syntax

`CnnPool='value'`

Parameter	Description
<code>value</code>	Values are: <ul style="list-style-type: none">• <code>Yes</code>• <code>No</code> (Default)

Default

No

Usage

If `CnnPool='Yes'`, the Oracle client creates a connection pool and can connect to Oracle9i, and Oracle 10g Server. The pool contains physical connections to Oracle Server and is managed by the OCI. The pool persists until the DataWindow server terminates or *OCI.dll* is unloaded.

The DataWindow server can connect to 10 different Oracle servers, as distinguished by service name, concurrently. There can be no more than 20 connections per pool. Each pool is created by the Oracle9i or later client. Once a connection pool has been created, the DataWindow server maintains the physical connections until it terminates or OCI.DLL is unloaded.

Examples

To use connection pooling, enter values in the following fields of the Connection page in the Database Profile Setup dialog box:

- *Profile Name* – Example value: `08i-pool`
- *Server* – Provide the full net service name created by Oracle Net. Example value: `adcora8i.appeon.com`
- *Login ID* – Example value: `scott`
- *Password*
- *Connect as* – Choose an item from the drop-down menu. Example value: `Normal`

Make sure Use Connection Pool is selected.

See also

`StatementCache`

CommitOnDisconnect

Description

Specifies whether InfoMaker should commit (the default) or roll back all previously uncommitted database updates before disconnecting from a data source.

When to specify CommitOnDisconnect

You must specify a value for CommitOnDisconnect *before* connecting to the database.

Controls

All database interfaces

Syntax

`CommitOnDisconnect='value'`

Parameter	Description
<i>value</i>	<p>Specifies whether InfoMaker should commit or roll back all previously uncommitted database updates before disconnecting from a data source. Values are:</p> <ul style="list-style-type: none"> • Yes (Default) InfoMaker commits all uncommitted database updates when an application closes. • No InfoMaker rolls back all uncommitted database updates when an application closes. With this setting, InfoMaker does not automatically commit updates when you disconnect from the database.

Default

`CommitOnDisconnect='Yes'`

Usage Set CommitOnDisconnect to No if you want InfoMaker to roll back uncommitted database updates (instead of automatically committing them when you disconnect from the database).

Examples To tell InfoMaker to roll back uncommitted database updates instead of committing them when disconnecting from the database, clear the Commit On Disconnect check box on the Connection page in the Database Profile Setup dialog box.

ConnectAs

Description Allows the user to connect to the Oracle Server with **SYSOPER** or **SYSDBA** system privileges. Supports proxy authentication with additional user names (ORA driver only) when connecting to Oracle 10.2 or higher database servers.

Controls O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12c

Syntax ConnectAs='value'

Parameter	Description
<i>value</i>	Values are: <ul style="list-style-type: none"><i>EndUser</i> (ORA driver only) Type the proxy user name you want to use for your connectionSYSDBASYSOPERNormal (Default)

Default Normal

Usage If ConnectAs='Normal', this parameter is not used. If ConnectAs ='SYSDBA' or ConnectAs ='SYSOPER', Oracle allows the user to connect with **SYSDBA** or **SYSOPER** privileges, respectively, provided that these privileges have been granted the user.

If you connect using ConnectAs ='SYSDBA', Oracle uses the **SYS** schema instead of the schema that might already be associated with the user ID. If you connect using ConnectAs ='SYSOPER', Oracle uses the **PUBLIC** schema.

The InfoMaker ORA driver supports the proxy authentication feature that was introduced in Oracle 10.2. With proxy authentication, the end user typically authenticates to a middle tier (such as a firewall), that in turn logs into the database on the user's behalf as a proxy user. After logging into the database, the proxy user can switch to the end user's identity and perform operations using the authorization accorded to that user.

The ConnectAs DBParm parameter allows you to take advantage of this proxy connection feature. For example, if the user's Transaction object LogID is "Scott" and you set the ConnectAs DBParm parameter to "John", the OCI client logs in to database as the proxy user ("Scott"), then switches to the end user identity ("John").

If you are using connection or session pooling, the proxy user name is the connection or session pooling creator (which you can provide in the PoolCreator and PoolPwd DBParm parameters), and the Transaction object's LogID is ignored. No proxy session can be created if pooling is set to homogeneous session mode.

Do not use with CnnPool='Yes'

Connection pooling cannot be used with this parameter. Do not select Use Connection Pool in the Database Profile Setup dialog box or set CnnPool to 'Yes'.

Examples

Example 1 To use the SYS schema instead of the schema associated with the User ID, select SYSDBA from the Connect As drop-down list on the Connection page in the Database Profile Setup dialog box, and for the 090 and 010 drivers, make sure Use Connection Pool is deselected.

Example 2 To use the proxy user schema instead of the schema associated with the User ID, type Scott in the Connect As drop-down list on the Connection page in the Database Profile Setup dialog box.

ConnectOption

Description

Sets driver-specific connection options when you are accessing an ODBC data source in InfoMaker. These options specify the following:

- How the ODBC driver prompts for additional connection information
- What type of security to use for a Microsoft SQL Server connection

- Whether the ODBC Driver Manager Trace is on or off and what trace file it uses
- Whether cursors are closed or left open on a **SQLTransact** call
- How temporary stored procedures are treated for a **SQLPrepare** call

Certain ConnectOption parameters apply to all ODBC drivers, whereas others apply only to particular ODBC drivers.

For information on each ConnectOption parameter and whether you can use it with your ODBC driver, see the table in the Syntax section.

When to specify ConnectOption

You must specify the ConnectOption parameter *before* connecting to an ODBC data source. The ConnectOption settings take effect when you connect to the database.

Controls

ODBC (if driver and back-end DBMS support this feature)

Syntax

```
ConnectOption=' SQL_DRIVER_CONNECT,value;  
SQL_INTEGRATED_SECURITY,value;  
SQL_OPT_TRACE,value;  
SQL_OPT_TRACEFILE,value;  
SQL_PRESERVE_CURSORS,value;  
SQL_USE_PROCEDURE_FOR_PREPARE,value '
```


The following table lists the applicable ODBC drivers, purpose, and values for each ConnectOption parameter.

Parameter	Description
SQL_DRIVER_CONNECT	<p>Driver Any ODBC driver that supports the <code>SQLDriverConnect</code> API call.</p> <p>Purpose Specifies how the ODBC driver prompts for additional connection information (such as the user ID and password) when connecting to an ODBC data source.</p> <p>Values The values you can specify are:</p> <ul style="list-style-type: none"> • SQL_DRIVER_COMPLETE (Default) If the connection string contains correct and sufficient information to connect, the driver connects to the specified data source. If any information is incorrect or missing, the driver displays one or more dialog boxes to prompt for the required connection parameters. The driver then connects to the specified data source. • SQL_DRIVER_COMPLETE_REQUIRED The driver takes the same actions as it does when <code>SQL_DRIVER_COMPLETE</code> is set. In addition, the driver disables the controls for any information not required to connect to the data source. • SQL_DRIVER_PROMPT The driver displays one or more dialog boxes to prompt for the required connection parameters. The driver then connects to the specified data source and builds a connection string from the information specified in the dialog boxes. • SQL_DRIVER_NOPROMPT If the connection string contains correct and sufficient information to connect, the driver connects to the specified data source. If any information is incorrect or missing, the driver returns an error.
SQL_INTEGRATED_SECURITY	<p>Driver Microsoft SQL Server ODBC driver (not supplied with InfoMaker).</p> <p>Purpose Specifies the type of connection to the Microsoft SQL Server database server.</p> <p>Values The values you can specify are:</p> <ul style="list-style-type: none"> • SQL_IS_OFF (Default) Request a normal (nontrusted) connection to SQL Server using standard security. If you specify <code>SQL_IS_OFF</code>, you cannot request a trusted connection to SQL Server using integrated security. • SQL_IS_ON Request a trusted connection to SQL Server using integrated security regardless of the login security currently in use on the database server. <p>For more about security mechanisms in Microsoft SQL Server, see the Microsoft documentation.</p>

Parameter	Description
SQL_OPT_TRACE	<p>Driver Any ODBC driver.</p> <p>Purpose Turns on or turns off the ODBC Driver Manager Trace in InfoMaker to troubleshoot a connection to an ODBC data source. The ODBC Driver Manager Trace provides detailed information about the ODBC API function calls that InfoMaker makes when connected to an ODBC data source.</p> <p>Values The values you can specify are:</p> <ul style="list-style-type: none"> • SQL_OPT_TRACE_OFF (Default) Turns off the ODBC Driver Manager Trace. • SQL_OPT_TRACE_ON Turns on the ODBC Driver Manager Trace. <p>For instructions on using the ODBC Driver Manager Trace, see “About ODBC Driver Manager” in <i>Connecting to Your Database</i>.</p>
SQL_OPT_TRACEFILE	<p>Driver Any ODBC driver.</p> <p>Purpose Specifies the name of the trace file where you want InfoMaker to send the output of the ODBC Driver Manager Trace. InfoMaker appends the output to the trace file you specify until you stop the trace. To display the trace file, you can use the File Editor (in InfoMaker) or any text editor (outside InfoMaker).</p> <p>Values You can specify any filename for the trace file, following the naming conventions of your operating system. By default, if tracing is on and you have not specified a trace file, InfoMaker sends ODBC Driver Manager Trace output to the file SQL.LOG.</p>
SQL_PRESERVE_CURSORS	<p>Driver Microsoft SQL Server ODBC driver (not supplied with InfoMaker).</p> <p>Purpose Specifies whether cursors are closed or left open on a SQLTransact call.</p> <p>Values The values you can specify are:</p> <ul style="list-style-type: none"> • SQL_PC_OFF (Default) Close all cursors on a SQLTransact call. • SQL_PC_ON Keep server cursors open on a SQLTransact call.

Parameter	Description
<code>SQL_USE_PROCEDURE_FOR_PREPARE</code>	<p>Driver Microsoft SQL Server ODBC driver (not supplied with InfoMaker).</p> <p>Purpose Specifies how temporary stored procedures are treated for a SQLPrepare call.</p> <p>Values The values you can specify are:</p> <ul style="list-style-type: none"> • SQL_UP_ON (Default) Generate temporary stored procedures for a SQLPrepare call. • SQL_UP_OFF Do not generate temporary stored procedures for a SQLPrepare call. The SQL statement is stored, compiled, and run at execution time. Syntax error checking does not occur until execution time. • SQL_UP_ON_DROP Explicitly drop temporary stored procedures for a subsequent SQLPrepare call or when a statement handle (hstmt) is freed for reuse.

Default `ConnectOption='SQL_DRIVER_CONNECT, SQL_DRIVER_COMPLETE; SQL_INTEGRATED_SECURITY,SQL_IS_OFF; SQL_OPT_TRACE,SQL_OPT_TRACE_OFF; SQL_PRESERVE_CURSORS,SQL_PC_OFF; SQL_USE_PROCEDURE_FOR_PREPARE,SQL_UP_ON'`

Usage *Microsoft SQL Server ODBC driver* The ConnectOption parameter applies only if you are accessing a SQL Server database with the Microsoft ODBC SQL Server driver.

You must obtain the Microsoft SQL Server ODBC driver from Microsoft Corporation. This driver is *not* supplied with InfoMaker.

Examples To specify nondefault options for the ConnectOption database parameter, complete the Options page in the Database Profile Setup - ODBC dialog box. Each ConnectOption parameter corresponds to an option in the dialog box, as follows:

ConnectOption parameter	Corresponding option
<code>SQL_DRIVER_CONNECT</code>	Connect Type
<code>SQL_INTEGRATED_SECURITY</code>	Integrated Security
<code>SQL_OPT_TRACE</code>	Trace ODBC API Calls
<code>SQL_OPT_TRACEFILE</code>	Trace File
<code>SQL_PRESERVE_CURSORS</code>	Preserve Cursors
<code>SQL_USE_PROCEDURE_FOR_PREPARE</code>	Use Procedure for Prepare

ConnectionString

Description

Specifies the parameters required to connect to an ODBC data source. InfoMaker uses these parameters to connect to the database.

Controls

ODBC

Syntax

The ConnectString syntax displays on a single line. You must enclose the entire ConnectString in single quotes and separate parameters within the ConnectString with semicolons.

```
ConnectString='DSN=data_source_name; {UID=user_ID;  
PWD=password; driver_specific_parameters'
```

Parameter	Description
<i>data_source_name</i>	A name that identifies the data source.
<i>user_ID</i>	(Optional) The user ID required to connect to the data source.
<i>password</i>	(Optional) The password required by <i>user_ID</i> to connect to the data source.
<i>driver_specific_parameters</i>	(Optional) Any other driver-specific parameters required to connect.

Default

None

Usage

InfoMaker generates the ConnectString automatically when you define an ODBC data source and copies it to the Preview box in the Database Profile Setup dialog box. This happens before you connect to the data source in InfoMaker.

Therefore, *you do not have to enter the ConnectString yourself* when defining an ODBC data source. However, you might need to edit the ConnectString value in the Database Profile Setup dialog box.

You can change the ConnectString parameter if necessary by editing it in the Database Profile Setup dialog box. For example, if you change the name of an existing ODBC data source, edit its database profile to update the connect string with the new DSN (data source name) value.

Examples

This example shows how to define a connect string for an ODBC data source that contains the data source name (DSN=Sales), user ID (UID=dba), and password (PWD=sql). On the Connection page in the Database Profile Setup dialog box, select Sales from the Data Source drop-down list, select the User ID check box and type *dba*, and select the Password check box and type *sql*.

CSIncr

Description Specifies an integer for database connection increments per session.

Controls ORA Oracle 11g/12c and Oracle 9.2 or later databases

Syntax CSIncr =value

Parameter	Description
value	Specifies an increment for the number of database connections in a connection or session pool.

Default 1

Usage This value is ignored when the SessionHomogeneous DBParm is set to false.

Examples To set the increment for the number of database connections in a session pool to 10, select Session Pooling from the Pooling Type drop-down list on the Pooling page in the Database Profile Setup dialog box, select the Homogeneous Session Pooling check box, and type 10 in the Increment text box on the same page.

See also CSMax
CSMin

CSMax

Description Specifies the maximum number of database connections in a connection or session pool.

Controls ORA Oracle 11g/12c and Oracle 9.2 or later databases

Syntax CSMax =value

Parameter	Description
value	Specifies the maximum number of database connections in a connection or session pool.

Default 100

Usage This value is ignored when the SessionHomogeneous DBParm is set to false.

Examples To set the maximum number of database connections in a connection pool to 15, select Connection Pooling from the Pooling Type drop-down list on the Pooling page in the Database Profile Setup dialog box, and type 15 in the Maximum Number of Sessions text box on the same page.

See also CSIncr
CSMin

CSMin

Description Specifies the minimum number of database connections in a connection or session pool.

Controls ORA Oracle 11g/12c and Oracle 9.2 or later databases

Syntax CSMin =*value*

Parameter	Description
<i>value</i>	Specifies the minimum number of database connections in a connection or session pool.

Default 1

Usage This value is ignored when the SessionHomogeneous DBParm is set to false.

Examples To set the minimum number of database connections in a connection pool to 5, select Connection Pooling from the Pooling Type drop-down list on the Pooling page in the Database Profile Setup dialog box, and type 5 in the Minimum Number of Sessions text box on the same page.

See also CSIncr
CSMax

CursorLib

Description Specifies the cursor library to use when connecting to an ODBC data source.

Controls ODBC (if driver and back-end DBMS support this feature)

Syntax CursorLib='*value*'

Parameter	Description
<i>value</i>	<p>The cursor library to use when connecting to an ODBC data source. Values are:</p> <ul style="list-style-type: none"> • ODBC_Cur_Lib Use the ODBC Version 2.0 or higher cursor library. • If_Needed Use the ODBC Version 2.0 or higher cursor library if your ODBC driver does not support cursors. • Driver_Cursors (Default) Use your data source's native cursor support.

Default CursorLib='Driver_Cursors'

Examples To specify use of the ODBC version 2.0 or later cursor library when connecting to an ODBC data source, select Cursor Library from the Cursor Library drop-down list on the Transaction page in the Database Profile Setup dialog box.

CursorLock

Description

When used with the CursorScroll parameter, specifies locking options for cursors in ODBC data source.

The values you can set for CursorLock control two aspects of cursor locking:

- **Concurrent access** Ensures that multiple users can simultaneously access data that is accurate and current.
- **Collision detection** Detects collisions that occur when multiple users update the same data at the same time.

Controls

ODBC (if driver and back-end DBMS support this feature)

Syntax

CursorLock='lock_value'

Parameter	Description
<i>lock_value</i>	<p>Specifies the type of locking you want to use for ODBC cursors. Values are:</p> <ul style="list-style-type: none"> • Lock Use the lowest level of locking sufficient to allow updates on table rows. • Opt Use optimistic concurrency control. This means that table rows are not locked against updates by other users. To detect collisions, compare row versions or timestamps. • OptVal Use optimistic concurrency control. This means that table rows are not locked against updates by other users. To detect collisions, compare selected values with their previous values. • ReadOnly Prohibit updates on table rows by any user. <p>For more about how the ODBC standard defines lock values, see your ODBC documentation.</p>

Default

If you do not specify a value for CursorLock, InfoMaker defaults to the cursor lock setting specified by your ODBC driver.

Examples

To set scrolling and locking options for cursors in an ODBC data source, select Dynamic Scrolling from the Scrolling Options drop-down list, and Optimistic Using Values from the Locking drop-down list on the Transaction page in the Database Profile Setup dialog box.

See also

[CursorScroll](#)

CursorScroll

Description

When used with the CursorLock parameter, specifies scrolling options for cursors in an ODBC data source.

The location of a cursor indicates the current position in the result set produced by a **SQL** statement. **Scrolling** allows a cursor to move through the data in a result set one row at a time.

Controls

ODBC (if driver and back-end DBMS support this feature)

Syntax

CursorScroll='scroll_value'

Parameter	Description
<i>scroll_value</i>	<p>Specifies the type of scrolling you want to use for ODBC cursors. Values are:</p> <ul style="list-style-type: none"> • Forward The cursor only scrolls forward through the result set. • Static The data in the result set does not change. • KeySet Specifies that the cursor is keyset-driven. When a keyset-driven cursor is opened, the driver saves keys for the <i>entire result set</i>. As the cursor scrolls through the result set, the driver uses the keys in this keyset to retrieve the current values for each row. • Dynamic The driver saves and uses only the keys for the rows specified in the rowset.

Default

If you do not specify a value for CursorScroll, InfoMaker defaults to the cursor scroll settings specified for your ODBC data source driver.

Usage

For large result sets, it might be impractical to use a keyset-driven cursor that requires the driver to save keys for the entire result set. Instead, you can use a **mixed cursor** by specifying a 32-bit integer value that is the number of rows in your keyset (see Example 2). This number is typically smaller than the result set. The default keyset size is 0.

A mixed cursor uses KeySet scrolling within the specified keyset and Dynamic scrolling outside the keyset.

Examples

Example 1 To set scrolling and locking options for cursors in an ODBC data source, on the Transaction page in the Database Profile Setup dialog box, select Dynamic Scrolling from the Scrolling Options drop-down list and Optimistic Using Values from the Locking drop-down list.

Example 2 This example sets the number of rows in the keyset to 100. Assume that the entire result set has 1000 rows. When the cursor is opened, the driver saves keys for the first 100 rows of the result set. It then retrieves the next block of 100 keys until the entire result set is retrieved.

Type **100** in the Scrolling Options box on the Transaction page in the Database Profile Setup dialog box.

See also

[CursorLock](#)

CursorType

Description

Supports the scrollable cursor feature introduced in Adaptive Server Enterprise 15.0, including directional scrolling (forwards and backwards) and sensitivity towards independent changes to table.

Controls

ASE, SYC SAP Adaptive Server Enterprise (15.0 and later)

Syntax

`CursorType='value'`

Parameter	Description
<i>value</i>	<p>A string that specifies whether database cursors are scrollable and whether they are sensitive to modifications in data. Values are:</p> <ul style="list-style-type: none"> • NonScrollable (Default) Non-scrollable forward-only cursor. Supports <code>FETCH NEXT</code> syntax. • ScrollInsensitive Scrollable insensitive cursor that ignores any data modifications when scrolling in either direction. Supports <code>FETCH NEXT</code>, <code>FETCH PRIOR</code>, <code>FETCH FIRST</code>, and <code>FETCH LAST</code> syntaxes. • ScrollSemiSensitive Scrollable semi-sensitive cursor that presents data modifications when scrolling forwards but ignores them when scrolling backwards. Supports <code>FETCH NEXT</code>, <code>FETCH PRIOR</code>, <code>FETCH FIRST</code>, and <code>FETCH LAST</code> syntaxes.

Default

`CursorType='NonScrollable'`

Usage

Adaptive Server Enterprise 15.0 allows both scrollable and nonscrollable cursors, which can be either semi-sensitive or insensitive. “Scrollable” means that you can scroll through the cursor result set by fetching any, or many, rows, rather than one row at a time; you can also scan the result set repeatedly. A scrollable cursor allows you to set the position of the cursor anywhere in the cursor result set for as long as the cursor is open.

To use a scrollable cursor, you must use a `DECLARE CursorName CURSOR SQL` statement to declare it with a suitable `SELECT` statement and you must have the query engine provided in Adaptive Server 15.0 or later.

For sensitive scrolling to work correctly, the table must have a clustered index or a clustered unique constraint, such as a clustered primary key.

All scrollable cursors are read-only and can only be used when the value of the CursorUpdate database parameter is 0 (the default). If you need an updatable cursor, set the CursorUpdate parameter to 1. When CursorUpdate is set to 1, the value of CursorType is ignored. All update cursors are nonscrollable.

Both client and server must be Adaptive Server 15.0 or higher.

Examples

To specify that database cursors support semi-sensitive scrolling (data modifications are presented when scrolling forwards), select Cursor Scrollable SemiSensitive from the Read Only Cursor Type drop-down list on the Transaction page in the Database Profile Setup dialog box.

See also

[CursorUpdate](#)

CursorUpdate

Description

For those interfaces that support it, CursorUpdate specifies whether cursors in your target database are declared read-only or updatable.

Controls

DIR SAP Sybase DirectConnect
ASE, SYC, SAP Adaptive Server Enterprise

Syntax

`CursorUpdate=value`

Parameter	Description
<i>value</i>	A number that specifies whether database cursors are declared read-only or updatable. Values are: <ul style="list-style-type: none">• 0 (Default) Cursors are declared read-only. SAP Sybase Client Library cursor declarations include the CS_READ_ONLY option.• 1 Cursors are declared updatable. SAP Sybase Client Library cursor declarations include the CS_FOR_UPDATE option. This option applies to all updatable columns in the table.

Default

CursorUpdate=0

Usage

Set the CursorUpdate parameter to 1 to declare updatable cursors if you plan to use either of the following `SQL` statements in your application (*table* represents the table name and *cursor* represents the cursor name):

```
DELETE FROM table WHERE CURRENT OF cursor  
UPDATE table SET set_clause WHERE CURRENT OF cursor
```

If you are communicating with the database in a InfoMaker script, you can reset the CursorUpdate value anytime before or after the Transaction object has connected to the database.

When you declare cursors updatable in a database accessed through DirectConnect, the cursor declaration you code must include a **FOR UPDATE OF *column_list*** clause.

When you use updatable cursors with the DIR interface and a Gatewayless connection to the mainframe, you must set Block=1 before executing the cursor. You can reset the Block parameter to its default of 100 after you close the cursor within your code.

Examples

To specify that database cursors are declared updatable, select the Cursors Declared Updatable check box on the Transaction page in the Database Profile Setup dialog box.

Database

Description

Specifies the name of the database you want to connect to.

When to specify Database

You must specify the Database parameter *before* connecting to the database.

Controls

SNC SQL Native Client for Microsoft SQL Server

Syntax

DataBase='*database_name*'

Default

None

Examples

To connect to the database “mydb”, enter mydb in the Database box on the Connection page in the Database Profile Setup dialog box.

DataLink

Description

Specifies that you want to create a file or use an existing file containing your connection information to connect to your data source.

When to specify DataLink

You must specify the DataLink parameter *before* connecting to the database.

Controls

OLE DB

SNC SQL Native Client for Microsoft SQL Server

Syntax

DataLink='file_name'

Default

None

Usage

The Data Link option allows you to access Microsoft's Data Link API. The Data Link API allows you to define a file or use an existing file that contains your connection information. A Data Link file is identified with the suffix *.udl*.

To launch the API, double-click on Manage Data Links under OLE DB Utilities in the Installed Database Interfaces list or select the File Name check box on the Connection page in the Database Profile Setup dialog box and click the button next to the File Name box.

For more information on using the Data Link API, see Microsoft's Universal Data Access Web site.

Using a Data Link file versus setting other database parameters If you use a Data Link file to connect to your data source with the OLE DB interface, all other settings you make in the Database Profile Setup dialog box are ignored.

If you use a Data Link file to connect to your data source with the SNC interface, the setting in the ProviderString database parameter still takes effect. The SNC interface gets a connection string from the data link file, and then copies the contents of the ProviderString parameter into the connection string so that it contains the connection parameters from both the data link file and the ProviderString parameter. You might want to take advantage of this feature if you do not want to save the user name and password in the UDL file. You can specify them in the ProviderString parameter instead.

Examples

To use the file *oledb.udl* to connect to an OLE DB data provider, select the File Name check box on the Connection page in the Database Profile Setup dialog box and enter a name for a new file or select an existing file.

DataSource

Description Identifies the data source to which you want to connect. The data source can be a file, a database, or an ODBC data source depending on the OLE DB data provider you are using.

When to specify DataSource

You must specify the DataSource parameter *before* connecting to the database.

Controls OLE DB

Syntax DataSource='datasource_name'

Default None

Usage For the **SNC** interface for **SQL Server 2005**, specifying a value for Server on the Connection page is equivalent to setting this parameter.

The value of the Data Source parameter varies depending on the type of data source connection you are making. For example, if you are using Microsoft's OLE DB Provider for ODBC, you would enter the actual ODBC data source name for the Data Source value. If you are using Microsoft's OLE DB Provider for **SQL Server**, you would enter the actual Microsoft **SQL Server** server name for the Data Source value.

For more information, see the documentation provided by your OLE DB data provider.

Examples **Example 1** To use the Microsoft OLE DB Provider for ODBC to connect to the PB Demo DB, enter **PB Demo DB** in the Data Source box on the Connection page in the Database Profile Setup dialog box.

Example 2 To use the Microsoft OLE DB Provider for Oracle to connect to an Oracle 12c database, enter the Oracle 12c server name in the Data Source box on the Connection page in the Database Profile Setup dialog box.

See also [DataLink Provider](#)

Date

Description When you update data in the Form painter, InfoMaker builds a **SQL UPDATE** statement in the background. The Date parameter determines how InfoMaker specifies a date datatype when it builds the **SQL UPDATE** statement.

Controls JDB JDBC

ODBC
 O90 Oracle9i
 O10 Oracle 10g
 ORA Oracle 11g/12c

Syntax

The syntax you use to specify the Date parameter differs slightly depending on the database.

The Database Profile Setup dialog box inserts special characters (quotes and backslashes) where needed, so you can specify just the date format.

JDBC and ODBC syntax InfoMaker parses the backslash followed by two single quotes (\ ' ') as a single quote when it builds the **SQL UPDATE** statement.

Date=' \"date_format\" '

Oracle syntax InfoMaker parses each set of four consecutive single quotes (' ' ' ') as a single quote when it builds the **SQL UPDATE** statement.

Date=' \"\"date_format\"\" '

Parameter	Description
' \"'	JDBC and ODBC syntax Type a single quote, followed by one space, followed by a backslash, followed by two single quotes. There is no space between the two single quotes and the beginning of the date format.
' \"\"'	Oracle syntax Type a single quote, followed by one space, followed by four single quotes. There is no space between the four single quotes and the beginning of the date format.
date_format	The date format you want InfoMaker to use when it builds a SQL UPDATE statement to update a data source in the Form painter. For more on display formats, see the <i>Users Guide</i> .
' \" ' '	JDBC and ODBC syntax Type a backslash, followed by two single quotes, followed by one space, followed by a single quote. There is no space between the end of the date format and the backslash.
' \"\" ' '	Oracle syntax Type four single quotes, followed by one space, followed by a single quote. There is no space between the end of the date format and the four single quotes.

Default

The default value for Date depends on the DBMS you are accessing, as summarized in the following table:

DBMS	Date default value
JDBC	If no value is specified for the Date database parameter, InfoMaker looks for a date format in the section for your JDBC driver in the registry. If no date format is found in the registry, InfoMaker uses the JDBC date format escape sequence.
ODBC	If no value is specified for the Date database parameter, InfoMaker looks for a date format in the section for your ODBC driver in the PBODB170 initialization file. If no date format is found in the initialization file, InfoMaker uses the ODBC date format escape sequence.
Oracle	The default Oracle date format. For information, see your Oracle documentation.

Examples

About these examples Assume you are updating a table named Employee by setting the Startdate column to 2006-04-23. This date is represented by the following date format:

```
yyyy-mm-dd
```

Example 1 (JDBC and ODBC syntax) To specify that InfoMaker should use this format for the date datatype when it builds the **SQL UPDATE** statement, type the following in the Date Format box on the Syntax page in the Database Profile Setup dialog box:

```
yyyy-mm-dd
```

What happens InfoMaker builds the following **SQL UPDATE** statement to update the table:

```
UPDATE EMPLOYEE
SET STARTDATE = '2006-04-23'
```

Example 2 (Oracle syntax) To specify that InfoMaker should use this format for the date datatype when it builds the **SQL UPDATE** statement, type the following in the Date format box on the Syntax page in the Database Profile Setup dialog box:

```
yyyy-mm-dd
```

What happens InfoMaker builds the following **SQL UPDATE** statement to update the table:

```
UPDATE EMPLOYEE
SET STARTDATE = '2006-04-23'
```

See also

[DateTime](#)

Time

DateFormat

Description

When you update data in the Form painter, InfoMaker builds a **SQL UPDATE** statement in the background. The DateFormat parameter determines how InfoMaker specifies a date datatype when it builds the **SQL UPDATE** statement.

Controls

OLE DB
SNC SQL Native Client for Microsoft SQL Server

Syntax

DateFormat='date_format'

Parameter	Description
<i>date_format</i>	The date format you want InfoMaker to use when it builds a SQL UPDATE statement to update a data source in the Form painter. For more on display formats, see the <i>Users Guide</i> .

Default

If no value is specified for the DateFormat parameter, InfoMaker does not use a date datatype.

Usage

When you call stored procedures, the database server might not accept the date format built by InfoMaker. If this occurs, you can try to use another format. For example, for Microsoft SQL Server, try this format:

```
DateFormat='\'\'yyyy-mm-dd\'\''
```

Examples

Assume you are updating a table named Employee by setting the Startdate column to 2006-04-23. This date is represented by the date format `yyyy-mm-dd`.

To specify that InfoMaker should use this format for the date datatype when it builds the **SQL UPDATE** statement, type the following in the Date Format box on the Syntax page in the Database Profile Setup dialog box:

```
yyyy-mm-dd
```

What happens InfoMaker builds the following **SQL UPDATE** statement to update the table:

```
UPDATE EMPLOYEE  
SET STARTDATE = '2006-04-23'
```

See also

DateTimeFormat
TimeFormat

DateTime

Description When you update data in the Form painter, InfoMaker builds a **SQL UPDATE** statement in the background. The DateTime parameter determines how InfoMaker specifies a DateTime datatype when it builds the **SQL UPDATE** statement. (A DateTime datatype contains both a date value and a time value.)

Controls JDB JDBC
ODBC
O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12c

Syntax The syntax you use to specify the DateTime differs slightly depending on the database.

The Database Profile Setup dialog box inserts special characters (quotes and backslashes) where needed, so you can specify just the DateTime format.

In code, you must use the following syntax:

JDBC and ODBC syntax InfoMaker parses the backslash followed by two single quotes (\") as a single quote when it builds the **SQL UPDATE** statement.

```
DateTime=' \"DateTime_format\" '
```

Oracle syntax InfoMaker parses each set of four consecutive single quotes (\"'\") as a single quote when it builds the **SQL UPDATE** statement.

```
DateTime=' \"'\DateTime_format\"'\\"'
```

Parameter	Description
'\"'	JDBC and ODBC syntax Type a single quote, followed by one space, followed by a backslash, followed by two single quotes. There is no space between the two single quotes and the beginning of the DateTime format.
'\"'\\"'	Oracle syntax Type a single quote, followed by one space, followed by four single quotes. There is no space between the four single quotes and the beginning of the date format.
<i>DateTime_format</i>	The DateTime format you want InfoMaker to use when it builds a SQL UPDATE statement to update a data source in the painter. For more on display formats, see the <i>Users Guide</i> .

Parameter	Description
<code>\"'</code>	JDBC and ODBC syntax Type a backslash, followed by two single quotes, followed by one space, followed by a single quote. There is no space between the end of the date format and the backslash.
<code>''''</code>	Oracle syntax Type four single quotes, followed by one space, followed by a single quote. There is no space between the end of the DateTime format and the four single quotes.

Default

The default value for DateTime depends on the DBMS you are accessing, as summarized in the following table:

DBMS	Date default value
JDBC	If no value is specified for the DateTime database parameter, InfoMaker looks for a DateTime format in the section for your JDBC driver in the registry. If no DateTime format is found in the registry, InfoMaker uses the JDBC DateTime format escape sequence.
ODBC	If no value is specified for the DateTime database parameter, InfoMaker looks for a DateTime format in the section for your ODBC driver in the <i>PBODBI70</i> initialization file. If no DateTime format is found in the initialization file, InfoMaker uses the ODBC DateTime format escape sequence.
Oracle	The default Oracle DateTime format. For information, see your Oracle documentation.

Examples

About these examples Assume you are updating a table named Files by setting the Timestamp column to 4/2/03 3:45 pm. This DateTime is represented by the following DateTime format.

Example 1 (ODBC and JDBC syntax) To specify that InfoMaker should use this format for the DateTime datatype when it builds the **SQL UPDATE** statement, type the following in the DateTime Format box on the Syntax page in the Database Profile Setup dialog box:

```
m/d/yy h:mm am/pm
```

What happens InfoMaker builds the following **SQL UPDATE** statement to update the table:

```
UPDATE FILES
SET TIMESTAMP = '4/2/03 3:45 pm'
```

Example 2 (Oracle syntax) To specify that InfoMaker should use this format for the DateTime datatype when it builds the **SQL UPDATE** statement, type the following in the DateTime Format box on the Syntax page in the Database Profile Setup dialog box:

m/d/yy h:mm am/pm

What happens InfoMaker builds the following SQL UPDATE statement to update the table:

```
UPDATE FILES
SET TIMESTAMP = '4/2/03 3:45 pm'
```

See also

Date
Time

DateTimeAllowed

Description

For those interfaces that support it, DateTimeAllowed controls whether columns having a DateTime datatype can appear as unique key columns in the WHERE clause of a SQL UPDATE or DELETE statement. InfoMaker generates an UPDATE statement or a DELETE statement followed by an INSERT statement to update the database from a form.

When you are working in the Form painter, you specify which columns to include in the WHERE clause by selecting them from the Unique Key Columns list in the Specify Update Properties dialog box.

By default, DateTimeAllowed is set to 0 to prohibit DateTime columns from displaying in the Unique Key Columns list and consequently from appearing in the WHERE clause of an UPDATE or DELETE statement. When you set DateTimeAllowed to 1, any DateTime columns in your database table display in the Unique Key Columns list and can be selected to appear in the WHERE clause of an UPDATE or DELETE statement.

When to specify DateTimeAllowed

You must specify a value for DateTimeAllowed *before* connecting to the database.

Controls

ASE, SYC SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect
I10 Informix
IN9 Informix

Syntax

DateTimeAllowed=*value*

Parameter	Description
<i>value</i>	<p>Specifies whether you can use DateTime columns as unique key columns in a WHERE clause of a SQL UPDATE or DELETE statement generated by InfoMaker to update the database. Values are:</p> <ul style="list-style-type: none"> • 0 (Default) Prohibit the use of DateTime columns in the WHERE clause of an UPDATE or DELETE statement. When DateTimeAllowed is set to 0, DateTime columns <i>do not display</i> in the Unique Key Columns list in the Specify Update Properties dialog box. You can also specify 'No' or 'False' to set this value. • 1 Allow the use of DateTime columns in the WHERE clause of an UPDATE or DELETE statement. When DateTimeAllowed is set to 1, DateTime columns <i>do display</i> in the Unique Key Columns list in the Specify Update Properties dialog box so you can select one or more to appear in the WHERE clause. You can also specify 'Yes' or 'True' to set this value.

Default

DateTimeAllowed=0

Usage

When to set To allow the use of DateTime columns as unique key columns in the **WHERE** clause of an **UPDATE** or **DELETE** statement when you are updating the database from a form, set DateTimeAllowed to 1.

For instructions on using the Specify Update Properties dialog box to specify update characteristics for a form, see the chapter on controlling updates in the *Users Guide*.

What happens when you save the form When you set DateTimeAllowed to 1, select a DateTime column to appear in the **WHERE** clause, and then save the form, this column continues to display in the Unique Key Columns list even if you set DateTimeAllowed to 0 on a subsequent connection.

Examples

To allow the use of DateTime columns in the **WHERE** clause of an **UPDATE** or **DELETE** statement, select the DateTime Datatype Allowed check box on the Syntax page in the Database Profile Setup dialog box.

DateTimeFormat

Description

When you update data in the Form painter, InfoMaker builds a **SQL UPDATE** statement in the background. The DateTimeFormat parameter determines how InfoMaker specifies a DateTime datatype when it builds the **SQL UPDATE** statement. (A DateTime datatype contains both a date value and a time value.)

Controls

OLE DB

SNC SQL Native Client for Microsoft SQL Server

Syntax

`DateTimeFormat='datetime_format'`

Parameter	Description
<code>datetime_format</code>	The datetime format you want InfoMaker to use when it builds a SQL UPDATE statement to update a data source in the Form painter. For more on display formats, see the <i>Users Guide</i> .

Default

If no value is specified for the `DateTimeFormat` parameter, InfoMaker does not use a datetime datatype.

Usage

When you call stored procedures, the database server might not accept the `DateTime` format built by InfoMaker. If this occurs, you can try to use another format. For example, for Microsoft **SQL Server**, try this format:

```
DateTimeFormat='\''yyyy-mm-dd hh:mm:ss.fff\'''
```

PowerBuilder parses the backslash followed by two single quotes (') as a single quote.

Examples

Assume you are updating a table named `Files` by setting the `Timestamp` column to 4/2/03 3:45 pm. This `DateTime` is represented by the following `DateTime` format.

To specify that InfoMaker should use this format for the `DateTime` datatype when it builds the **SQL UPDATE** statement, type the following in the `DateTime Format` box on the `Syntax` page in the `Database Profile Setup` dialog box:

```
m/d/yy h:mm am/pm
```

What happens InfoMaker builds the following **SQL UPDATE** statement to update the table:

```
UPDATE FILES
SET TIMESTAMP = '4/2/03 3:45 pm'
```

See also

`DateFormat`
`TimeFormat`

DBGetTime

Description

Specifies the number of seconds InfoMaker waits for a response from the DBMS when you retrieve rows in a report, form, or query. When you set the Async parameter to 1 to enable asynchronous operation, you can also set the DBGetTime parameter for those DBMSs that support this parameter.

If DBGetTime is set to 0 (the default), InfoMaker waits indefinitely for a DBMS response (the request never times out). If the DBGetTime value expires before the first row is retrieved, your request is automatically canceled.

Controls

ASE, SYC SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect
I10 Informix
IN9 Informix
JDB JDBC
ODBC (if driver and back-end DBMS support this feature)
O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12c

Syntax

DBGetTime=*value*

Parameter	Description
<i>value</i>	The number of seconds InfoMaker waits for a DBMS response while waiting to retrieve the first row of a report or form.

Default

DBGetTime=0

Usage

Requirements for using DBGetTime To use the DBGetTime parameter, you must do *both* of the following:

- Set the Async parameter to 1 to enable asynchronous operation, as shown in the Examples.
- Code a RetrieveRow event for a form or report.

Examples

To enable asynchronous operation and set the DBGetTime parameter to 20 seconds, select the Asynchronous check box and type 20 in the Number Of Seconds To Wait box on the Transaction page in the Database Profile Setup dialog box.

See also

[Async](#)

Db_Locale

Description

Db_Locale identifies the locale of the data in the database.

When to specify Db_Locale

You must specify the Client_Locale parameter *before* connecting to a database.

Controls

I10 Informix

Syntax

Db_Locale='language_territory.codeset'

Parameter	Description
<i>language</i>	Two character name that represents the language for a specific locale, for example “en” for English.
<i>territory</i>	Two character name that represents the cultural conventions for a specific territory, for example “AU” for Australia.
<i>codeset</i>	Name of the code set that the locale supports, for example “utf8.”

Default

Client_Locale value.

Usage

The I10 native interface uses the Informix GLS (Global Language Support) API for global language support. Db_Locale specifies the value of the Informix environment variable DB_LOCALE. If you do not set the DbParameter, the default Db_Locale value is the Client_Locale value.

Db_Locale specifies the language, territory, and code set that the database server needs to correctly interpret locale-sensitive datatypes such as **NChar** and **NVarChar** in a specific database. The code set specified in DB_LOCALE determines which characters are valid in any character column, as well as in the names of database objects such as databases, tables, columns, and views.

For more information about the Informix CLIENT_LOCALE and DB_LOCALE environment variables, see the *IBM Informix GLS User's Guide*, currently available at the Informix library web site.

Examples

To set the character set to en_us.utf8, type `en_us.utf8` in the Database Locale box on the Regional Settings page in the Database Profile Setup dialog box.

See also

[Client_Locale](#)
[Locale](#)

DBTextLimit

Description

Specifies the maximum length of a text field that is returned when you include the text field in a **SQL SELECT** statement.

You can set the DBTextLimit parameter if you want to include a long text string in a report without treating the text as a binary large object (blob) datatype.

Controls

ASE, SYC, SAP Adaptive Server Enterprise
 DIR SAP Sybase DirectConnect
SNC SQL Native Client for Microsoft **SQL** Server

Syntax

DBTextLimit='value'

Parameter	Description
<i>value</i>	The maximum length in bytes of a text field that is returned when you include the text field in a SQL SELECT statement. The range of valid values is from 0 bytes to 2 GB. When you set DBTextLimit to 0, the server returns the maximum length text field.

Default

The default value for DBTextLimit is the default specified for the DBTEXTLIMIT DB-Library or CS_TEXTLIMIT CT-Library connection property.

Usage

The text field length that DB-Library or CT-Library returns is the lesser of the DBTextLimit value and the setting for the global variable TEXTSIZE.

If the setting for TEXTSIZE is less than the value you specify for DBTextLimit, DB-Library or CT-Library returns the TEXTSIZE value.

Examples

To have DB-Library or CT-Library return a text field that is up to 32,000 bytes long when you include the text field in a **SQL SELECT** statement, type **32000** in the Text Limit in **SQL** box (when using the ASE or SYC interface), or Maximum Length of Long VarChar box (when using the DirectConnect interface) on the Syntax page in the Database Profile Setup dialog box.

DecimalSeparator

Description

Specifies the decimal separator setting used by the back-end DBMS that you are accessing in InfoMaker. If your DBMS uses a decimal separator other than period (.), which is the default, set DecimalSeparator to the value for your DBMS to ensure that InfoMaker correctly handles numeric strings returned from your database.

Controls

DIR SAP Sybase DirectConnect

ODBC (if driver and back-end DBMS support this feature)
 OLE DB
 O90 Oracle9i
 O10 Oracle 10g
 ORA Oracle 11g/12c
 SNC SQL Native Client for Microsoft SQL Server

Syntax

`DecimalSeparator='value'`

Parameter	Description
<i>value</i>	<p>The decimal separator setting used by the back-end DBMS that you are accessing in InfoMaker. Values are:</p> <ul style="list-style-type: none"> '.' (Default) Specifies that your back-end DBMS uses a period (.) as the decimal separator. If you do not specify <code>DecimalSeparator</code> or if you specify a value other than period (.) or comma (,), InfoMaker uses period (.) as the decimal separator. ',' Specifies that your back-end DBMS uses a comma (,) as the decimal separator.

Default

`DecimalSeparator='.'`

Usage

When to set `DecimalSeparator` The `DecimalSeparator` parameter currently supports period (.) and comma (,) as valid values. Therefore, if the decimal separator setting for your DBMS is a comma, you should set the `DecimalSeparator` parameter to ',' (comma) to make sure InfoMaker correctly handles numeric strings returned from your database.

Example using Oracle Assume you are accessing an Oracle database in InfoMaker and the decimal separator setting is a comma (,). Oracle returns to InfoMaker the numeric string '123,50' containing a comma instead of a period as the decimal separator. InfoMaker then sends this string to its decimal conversion routines.

By default, the InfoMaker decimal conversion routines expect a period as the decimal separator. If you set the `DecimalSeparator` parameter to ',' (comma), InfoMaker correctly handles this string and returns it as '123,50'.

Examples

To specify that your DBMS uses a comma (,) as the decimal separator setting, type a comma (,) in the `Decimal Separator` box on the Syntax page in the Database Profile Setup dialog box.

See also

[NumericFormat](#)

DefaultProcOwner

Description

The DefaultProcOwner parameter lets you set a default owner for a stored procedure. The parameter takes effect only when the stored procedure is not qualified. For ODBC, the PBNewSPInvocation parameter must also be set.

Controls

ODBC

Syntax

DefaultProcOwner=*'value'*

Parameter	Description
<i>value</i>	A string specifying the name of the default owner of the stored procedure

Usage

The parameter can be set dynamically at runtime after connecting to a database. You can also set it in your *PBODBI70.INI* file if you want to create and retrieve data into a report with a stored procedure data source in the Report painter. The runtime setting overrides the setting in *PBODBI70.INI*.

You can also cancel the setting at runtime. If you do so, InfoMaker uses the current user as the owner of a non-qualified stored procedure when it obtains the parameters of the stored procedure.

Examples

To set the default owner to `proms` in *PBODBI70.INI*:

```
[SQL Anywhere]
DefaultProcOwner='proms'
```

See also

PBNewSPInvocation

DelimitIdentifier

Description

Specifies whether you want InfoMaker to enclose the names of tables, columns, indexes, and constraints in double quotes when it generates **SQL** statements. This affects the behavior of any InfoMaker painter that generates **SQL** syntax.

Controls

ASE, SYC SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect
I10 Informix
IN9 Informix
JDB JDBC
ODBC (if driver and back-end DBMS support this feature)
OLE DB
O90 Oracle9i

O10 Oracle 10g

ORA Oracle 11g/12c

SNC SQL Native Client for Microsoft SQL Server

Syntax

`DelimitIdentifier='value'`

Parameter	Description
<i>value</i>	Specifies whether you want InfoMaker to enclose table and column names in double quotes. Values are: <ul style="list-style-type: none"> • Yes Use double quotes • No Do not use double quotes

Default

The default value for the DelimitIdentifier parameter depends on the DBMS you are accessing, as follows:

DBMS	DelimitIdentifier default value
Informix	DelimitIdentifier = 'No'
JDBC	Depends on the DelimitIdentifier setting in the registry
ODBC	Depends on the DelimitIdentifier setting in the PBODB170 initialization file
Oracle	DelimitIdentifier = 'Yes'
OLE DB	Depends on the DelimitIdentifier setting in the PBODB170 initialization file
SNC	DelimitIdentifier = 'No'
SAP Sybase DirectConnect	DelimitIdentifier = 'No'
SAP Adaptive Server Enterprise	DelimitIdentifier = 'No'

Usage

Informix Informix database servers can create a log of database transactions in either ASCII or non-ASCII format. If the database is creating a non-ASCII log, the setting of the DelimitIdentifier is optional. If the database is creating an ASCII log, you must set DelimitIdentifier='Yes' to make the SQL syntax generated by InfoMaker behave as expected.

SAP Adaptive Server Enterprise When you set DelimitIdentifier to 'Yes', the “set quoted_identifier on” command is automatically sent to Adaptive Server to adjust your database connection on the server. Otherwise, the “set quoted_identifier off” command is sent to the server. This feature occurs with ASE, SYC, JDBC, ODBC, and OLE DB interfaces.

Microsoft SQL Server When you set DelimitIdentifier to 'Yes', the “set quoted_identifier on” command is automatically sent to Microsoft **SQL** Server to adjust your database connection on the server when you use ODBC or OLE DB. Otherwise, the “set quoted_identifier off” command is sent to the server.

Sending “set quoted_identifier off” to the server can cause some SQL commands to fail if the SQL code contains quotation marks. To prevent InfoMaker from sending this instruction to the server, set the DelimitIdentifierToDB parameter to 'No' in the PBODB initialization file or the connection string.

JDBC and ODBC The DelimitIdentifier parameter setting overrides the DelimitIdentifier setting specified for your JDBC driver in the registry and for your ODBC driver in the PBODB170 initialization file.

DirectConnect If you want to use mixed-case identifier names, you must set DelimitIdentifier='Yes'. Also, you must set LowerCaseIdent='No' to preserve case sensitivity of identifiers stored in the DB2 system catalog.

Examples

To specify that InfoMaker should not enclose table and column names in double quotes when it generates **SQL** statements, clear the Enclose Table And Column Names In Quotes check box on the Syntax page in the Database Profile Setup dialog box.

See also

[DelimitIdentifierToDB](#)
[LowerCaseIdent](#)

DelimitIdentifierToDB

Description

Specifies whether InfoMaker should send a “set quoted_identifier off” instruction to the server when the DelimitIdentifier parameter is not set.

Controls

ODBC (if driver and back-end DBMS support this feature)
OLE DB

Syntax

DelimitIdentifierToDB='value'

Parameter	Description
value	Specifies whether InfoMaker sends a “set quoted_identifier off” instruction to the server when the DelimitIdentifier parameter is not set. Values are: <ul style="list-style-type: none">• Yes Send set quoted_identifier off to the server• No Do not send set quoted_identifier off to the server

Default

'No'

Usage When you set `DelimitIdentifier` to 'Yes', the "set quoted_identifier on" command is automatically sent to the database server to adjust your database connection on the server when you use ODBC or OLE DB. Otherwise, the "set quoted_identifier off" command is sent to the server.

Sending "set quoted_identifier off" to the server can cause some **SQL** commands to fail if the **SQL** code contains quotation marks. To prevent InfoMaker from sending this instruction to the server, set the `DelimitIdentifierToDB` parameter to 'No' in the PBODB initialization file or the connection string.

Examples To specify that InfoMaker should not send a set quoted_identifier instruction to the server, add the following to the appropriate section of your PBODB initialization file:

```
"DelimitIdentifierToDB='No'"
```

See also [DelimitIdentifier](#)

DisableBind

Description For those DBMSs that support bind variables, InfoMaker can bind input parameters to a compiled **SQL** statement by default. The `DisableBind` parameter allows you to specify whether you want to disable this binding.

When you set `DisableBind` to 1 to disable the binding, InfoMaker replaces the input variable with the value entered by the application user or specified in code.

Controls ASE, SYC SAP Adaptive Server Enterprise
 I10 Informix
 IN9 Informix
 JDB JDBC
 ODBC (if driver and back-end DBMS support this feature)
 OLE DB
 O90 Oracle9*i*
 O10 Oracle 10*g*
 ORA Oracle 11*g*/12*c*
 SNC **SQL** Native Client for Microsoft **SQL** Server

Syntax `DisableBind=value`

Parameter	Description
<i>value</i>	<p>Specifies whether you want to disable the binding of input parameters to a compiled SQL statement. Values are:</p> <ul style="list-style-type: none"> • 0 InfoMaker binds input parameters to a compiled SQL statement. • 1 InfoMaker does <i>not</i> bind input parameters to a compiled SQL statement.

Default

DisableBind=1 for ASE, SYC, SNC and OLE DB, DisableBind=0 for other interfaces

Usage

Bind variables In a SQL statement, a **bind variable** is a placeholder for a column value. By default, InfoMaker associates (binds) data from a variable defined in your application to the bind variable each time the SQL statement executes.

Using bind variables in SQL statements For example, the following SQL statement retrieves those rows in the Books table about books written by Hemingway:

```
SELECT * FROM books WHERE author="Hemingway"
```

Suppose that you want to execute this statement to get information about books written by other authors. Instead of compiling and executing a new statement for each author, you can define a bind variable that represents the author's name. The user then supplies the author's actual name when the application executes. By using bind variables, you ensure that the statement is compiled only once and executed repeatedly with new values supplied by the user.

If your database supports bind variables and DisableBind is set to 0 to enable binding (the default for all database interfaces except ASE, SYC, SNC and OLE DB), InfoMaker generates the statement with parameter markers (:bind_param) and passes the actual parameter value at execution time. For example:

```
SELECT * FROM books WHERE author=:bind_param
```

Bind variables and cached statements Using bind variables in conjunction with cached statements can improve the performance of most applications, depending on the application. In general, applications that perform a large amount of transaction processing benefit the most from using bind variables and cached statements.

In order to use cached statements, make sure that `DisableBind` is set to 0. This enables the binding of input variables to **SQL** statements in InfoMaker. (For more about using cached statements, see the description of the **SQLCache** parameter.)

Performance improvements For **SQL** Anywhere and Oracle databases, bind variables improve performance by allowing InfoMaker to insert and modify strings that exceed 255 characters.

Bind variables and default column values When `DisableBind` is set to 0 to enable the use of bind variables, the Form painter does both of the following to get maximum performance improvement from using bind variables when you add rows to a form:

- Generates a **SQL INSERT** statement that includes all columns (except identity and **SQL** Server timestamp)
- Reuses this **SQL INSERT** statement for each row you add to the form

For example, if a table named `Order_T` contains three columns named `Order_ID`, `Order_Date`, and `Customer_ID`, the Form painter generates the following **SQL INSERT** statement when `DisableBind` is set to 0 (default binding enabled):

```
INSERT INTO Order_T(Order_ID, Order_Date, Customer_ID)
VALUES (:bind_param1, :bind_param2, :bind_param3)
```

If one of these columns is null, the Form painter sets a null value indicator for this column parameter and executes the statement. This behavior is important to understand if you want your back-end DBMS to set a default value for any columns in your form.

To illustrate, suppose that your application users do not enter a value for the `Order_Date` column because they expect the back-end DBMS to set this column to a default value of `TODAY`. Then, they retrieve the row and find that a null value has been set for `Order_Date` instead of its default value. This happens because the **SQL INSERT** statement generated by the Form painter specified a null value indicator, so the DBMS set the column value to null instead of to its default value as expected.

Setting a default column value when binding is enabled If you are using bind variables (`DisableBind` set to 0) and want the back-end DBMS to set a column to its default value when your application user does not explicitly enter a value in a new row, you should set an initial value for the form column that mirrors the DBMS default value for this column.

In the Form painter, you can set or modify a column's initial value in the Column Specifications dialog box.

For more about the Column Specifications dialog box, see the *Users Guide*.

Setting a default column value when binding is disabled If you are *not* using bind variables (DisableBind set to 1) and want the back-end DBMS to set a column to its default value when your application user does not explicitly enter a value in a new row, you do *not* need to set an initial value for the form column.

This is because with bind variables disabled, the Form painter generates a **SQL INSERT** statement for each row added to the form. If a column does not contain an explicit value, it is not included in the **SQL INSERT** statement.

Using the Order_T table example, if your application user enters 123 as the value for the Order_ID column and A-123 as the value for the Customer_ID column, the Form painter generates the following **SQL INSERT** statement when DisableBind is set to 1 (binding disabled):

```
INSERT INTO Order_T(Order_ID, Customer_ID)
VALUES (123, 'A-123')
```

Your back-end DBMS would then set the Order_Date column to its default value as expected, since a value for Order_Date is not explicitly set in the **SQL INSERT** statement generated by the Form painter.

Examples

To specify that InfoMaker should disable the binding of input parameters to a compiled **SQL** statement, select the Disable Bind check box on the Transaction page in the Database Profile Setup dialog box.

See also

[SQLCache](#)

Driver

Description

The JDBC driver your application uses to connect to the database.

When to specify Driver

You must specify the Driver database parameter *before* connecting to the database.

Controls

JDB JDBC

Syntax

Driver='driver_name'

Default

None

Usage

The driver name identifies the Java class name for the particular driver you are using to connect to the database.

Examples

Example 1 To set the driver name of an SAP Sybase jConnect driver, type the following in the Driver Name box on the Connection page in the Database Profile Setup dialog box:

```
com.sybase.jdbc.SybDriver
```

Example 2 To set the driver name of an Oracle JDBC Driver, type the following in the Driver Name box on the Connection page in the Database Profile Setup dialog box.

```
oracle.jdbc.driver.OracleDriver
```

See also

[URL](#)

DS_Alias

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, DS_Alias is one of several parameters that you can set to enable network-based directory services in your application. (For other directory services parameters, see the See Also section.)

Some directory service providers and drivers support the creation of alias entries. An **alias entry** provides a link to a primary directory entry in a hierarchy, thereby giving users multiple ways to access the primary entry while searching the directory structure for a particular network entity.

For those directory service providers and drivers that support aliases, DS_Alias specifies whether the provider is allowed to follow links for (expand) alias entries while searching the directory hierarchy. The default behavior is to allow expansion of alias entries for providers that support this feature.

You must specify a value for DS_Alias *before* connecting to the database in InfoMaker.

Using third-party directory service providers

For information about the third-party directory service providers and operating system platforms that Apeon has tested with Open Client directory services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

DS_Alias=*value*

Parameter	Description
<i>value</i>	<p>For those directory services providers and drivers that support aliases, specifies whether the provider is allowed to expand alias entries while searching a directory hierarchy. Values are:</p> <ul style="list-style-type: none"> • 0 Prohibit provider from expanding alias entries during a directory search. You can also specify 'No' or 'False' to set this value. • 1 (Default) Allow provider to expand alias entries during a directory search. You can also specify 'Yes' or 'True' to set this value.

Default

DS_Alias=1

Usage

When to use To prevent access to your data through directory alias entries, set DS_Alias to 0. This prohibits directory service providers that support aliases from expanding alias entries during a directory search.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the [Release](#) parameter for more information.

Requirements for use To use DS_Alias or any other parameter supporting Open Client directory services, you must meet certain requirements for using directory services in your InfoMaker application. For details, see “Requirements for using Open Client directory services” in [Connecting to Your Database](#).

Corresponding CT-Lib connection property Specifying a value for DS_Alias sets the corresponding SAP Sybase CT-Lib connection property named CS_DS_EXPANDALIAS.

Examples

To prohibit directory service providers that support aliases from expanding alias entries during a directory search:

- **Database profile** Clear the Directory Alias Entries check box on the Directory Services page in the Database Profile Setup dialog box.

See also

DS_Copy
 DS_DitBase
 DS_Failover
 DS_Password
 DS_Principal
 DS_Provider
 DS_TimeLimit
 Release

DS_Copy

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, DS_Copy is one of several parameters that you can set to enable network-based directory services in your application. (For other directory services parameters, see the See Also section.)

Some directory service providers and drivers support the use of caching. **Caching** allows a directory service provider to use cached information while searching a directory instead of making a request to the directory server agent for information.

For those directory service providers and drives that support caching, DS_Copy specifies whether the provider is allowed to use cached information during a directory search. The default behavior is to allow providers that support this feature to use cached information.

You must specify a value for DS_Copy *before* connecting to the database in InfoMaker.

Using third-party directory service providers

For information about the third-party directory service providers and operating system platforms that Apeon has tested with Open Client directory services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

DS_Copy=*value*

Parameter	Description
<i>value</i>	<p>For those directory services providers and drivers that support caching, specifies whether the provider is allowed to use cached information when making a directory search. Values are:</p> <ul style="list-style-type: none"> • 0 Prohibit provider from using cached information during a directory search. You can also specify 'No' or 'False' to set this value. • 1 (Default) Allow provider to use cached information when making a directory search. You can also specify 'Yes' or 'True' to set this value.

Default

DS_Copy=1

Usage

When to use Allowing providers to use cached information during directory searches makes the searches faster, but does not ensure that the provider is using the most up-to-date directory information.

To ensure that the application gets the most recent changes to directory entries when it requests directory information, set DS_Copy to 0 to prohibit providers that support caching from using cached information during a directory search.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the *Release* parameter for more information.

Requirements for use To use DS_Copy or any other parameter supporting Open Client directory services, you must meet certain requirements for using directory services in your InfoMaker application. For details, see “Requirements for using Open Client directory services” in *Connecting to Your Database*.

Corresponding CT-Lib connection property Specifying a value for DS_Copy sets the corresponding SAP Sybase CT-Lib connection property named CS_DS_COPY.

Examples

To prohibit directory service providers that support caching from using cached information during a directory search:

- **Database profile** Clear the Use Caching check box on the Directory Services page in the Database Profile Setup dialog box.

See also

DS_Alias
DS_DitBase
DS_Failover
DS_Password
DS_Principal
DS_Provider
DS_TimeLimit
Release

DS_DitBase

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, DS_DitBase is one of several parameters that you can set to enable network-based directory services in your application. (For other directory services parameters, see the See Also section.)

When you use Open Client directory services, a default (active) directory information tree base (DIT base) is specified in the Open Client/Server™ Configuration utility. The **DIT base** is the directory node where directory searches start. This is analogous to the current working directory in MS-DOS file systems.

DS_DitBase lets you specify the name of the directory node where you want searches for directory entries to start. The DS_DitBase value you specify must be a fully qualified name that uses the syntax required by your directory service provider and driver (see the Examples section for illustrations).

The default value for DS_DitBase is the DIT base currently specified as active in the Open Client/Open Server Configuration utility.

You must specify a value for DS_DitBase *before* connecting to the database in InfoMaker.

Using third-party directory service providers

For information about the third-party directory service providers and operating system platforms that Apeon has tested with Open Client directory services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

DS_DitBase=*dit_base*

Parameter	Description
<i>dit_base</i>	<p>The name of the directory node where you want directory searches to start. By default, this is the DIT base currently specified as active in the Open Client/Open Server Configuration utility.</p> <p>The value for <i>dit_base</i> must be a fully qualified name that uses the syntax required by your directory service provider and driver. The syntax for specifying the DIT base varies for different providers; see your provider's documentation for details.</p> <p>For examples of how to specify <i>dit_base</i> for different directory service providers, see the Examples section.</p>

Default The default value for DS_DitBase is the DIT base currently specified as active in the Open Client/Open Server Configuration utility.

Usage *When to use* Set DS_DitBase to specify a starting node for directory searches *other than* the DIT base node specified as active in the Open Client/Open Server Configuration utility. For instructions on using the Open Client/Open Server Configuration utility, see your SAP Sybase Open Client/Server configuration guide.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the **Release** parameter for more information.

Requirements for use To use DS_DitBase or any other parameter supporting Open Client directory services, you must meet certain requirements for using directory services in your InfoMaker application. For details, see “Requirements for using Open Client directory services” in *Connecting to Your Database*.

Corresponding CT-Lib connection property Specifying a value for DS_DitBase sets the corresponding SAP Sybase CT-Lib connection property named CS_DS_DITBASE.

Examples **About these examples** The examples that follow show how to specify a DS_DitBase value for different directory service providers.

See your directory service provider’s documentation for complete information about the format your provider requires for specifying the DIT base.

Example 1 (Windows NT Registry) This example shows the syntax for DS_DitBase if your directory service provider is the Windows NT Registry:

```
Node name: SALES:software\sap\server\SYS11NT
DS_DitBase: SALES:software\sap\server
```

To set DS_DitBase:

- **Database profile** Type the following in the DIT Base box on the Directory Services page in the Database Profile Setup dialog box. Do *not* end the DS_DitBase value with a backslash (\):

```
SALES:software\sybase\server
```

Example 2 (DCE/CDS) This example shows the syntax for DS_DitBase if your directory service provider is Distributed Computing Environment Cell Directory Services (DCE/CDS):

```
Node name: /.../boston.sales/dataservers/sap/SYS11
DS_DitBase: /.../boston.sales/dataservers
```

To set DS_DitBase:

- **Database profile** Type the following in the DIT Base box on the Directory Services page in the Database Profile Setup dialog box. Do *not* end the DS_DitBase value with a slash (/):

```
/.../boston.sales/dataservers
```

Example 3 (Novell NDS) This example shows the syntax for DS_DitBase if your directory service provider is Novell NetWare Directory Services (NDS):

```
Node name: CN=SYS11.OU=miami.OU=sales.O=sap
DS_DitBase: OU=miami.OU=sales.O=sap
```

To set DS_DitBase:

- **Database profile** Type the following in the DIT Base box on the Directory Services page in the Database Profile Setup dialog box:

```
OU=miami.OU=sales.O=sap
```

See also

DS_Alias
DS_Copy
DS_Failover
DS_Password
DS_Principal
DS_Provider
DS_TimeLimit
Release

DS_Failover

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, DS_Failover is one of several parameters that you can set to enable network-based directory services in your application. (For other directory services parameters, see the See Also section.)

SAP Sybase Open Client Client-Library (CT-Lib) requires a directory to map logical server names to network addresses. The source for this directory can be either the SAP Sybase Interfaces file or a network-based directory service provider (such as DCE Cell Directory Services or the Windows Registry).

If you want an application to use a directory source *other than* the Interfaces file, CT-Lib must be able to load the appropriate directory driver. If CT-Lib cannot load the required driver, you can set DS_Failover to specify whether CT-Lib should silently default (fail over) to using the Interfaces file as the directory source.

By default, DS_Failover specifies that CT-Lib should use the Interfaces file as the directory source if it cannot load the requested directory driver.

You must specify a value for DS_Failover *before* connecting to the database in InfoMaker.

Using third-party directory service providers

For information about the third-party directory service providers and operating system platforms that Appeon has tested with Open Client directory services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

DS_Failover=*value*

Parameter	Description
<i>value</i>	Specifies whether SAP Sybase CT-Lib should silently default (fail over) to using the Interfaces file as the directory source if it cannot load the requested directory driver. Values are: <ul style="list-style-type: none">• 0 Prohibit CT-Lib from using the Interfaces file as the directory source if it cannot load the requested directory driver. You can also specify 'No' or 'False' to set this value.• 1 (Default) Allow CT-Lib to use the Interfaces file as the directory source if it cannot load the requested directory driver. You can also specify 'Yes' or 'True' to set this value.

Default

DS_Failover=1

Usage

When to use To prevent CT-Lib from using the Interfaces file as the directory source if it cannot load the requested directory driver, set DS_Failover to 0.

If DS_Failover is set to 0 to prevent use of the Interfaces file and CT-Lib cannot load the requested directory driver, the connection's directory source is undefined. This causes certain operations requiring directory access to fail.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the [Release](#) parameter for more information.

Requirements for use To use DS_Failover or any other parameter supporting Open Client directory services, you must meet certain requirements for using directory services in your InfoMaker application. For details, see “Requirements for using Open Client directory services” in *Connecting to Your Database*.

Corresponding CT-Lib connection property Specifying a value for DS_Failover sets the corresponding SAP Sybase CT-Lib connection property named CS_DS_FAILOVER.

Examples

To prohibit CT-Lib from using the Interfaces file as the directory source if it cannot load the requested directory driver:

- **Database profile** Clear the Enable Failover check box on the Directory Services page in the Database Profile Setup dialog box.

See also

[DS_Alias](#)
[DS_Copy](#)
[DS_DitBase](#)
[DS_Password](#)
[DS_Principal](#)
[DS_Provider](#)
[DS_TimeLimit](#)
[Release](#)

DS_Password

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client 12.5 or higher software, DS_Password is one of several parameters that you can set to enable network-based directory services in your application. (For other directory services parameters, see the See Also section.)

Some directory service providers and drivers require an authenticated principal (user ID) name and password to control an application's access to directory entries. For those providers and drivers, DS_Principal and DS_Password specify the principal name and password your application should use to identify you to the directory service provider.

You must specify a value for DS_Password *before* connecting to the database in InfoMaker.

Controls ASE, SYC SAP Adaptive Server Enterprise

Syntax DS_Password='password'

Parameter	Description
<i>password</i>	The password associated with the principal (user ID) name you specified in the DS_Principal parameter.

Default None

InfoMaker does not set DS_Password or the corresponding SAP Sybase Open Client Client-Library (CT-Lib) connection parameter CS_DS_PASSWORD if you do not specify a value.

Usage *When to use* If your directory service provider requires an authenticated principal name for directory access, set DS_Password to the password that goes with your directory service principal name.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the [Release](#) parameter for more information.

Requirements for use To use DS_Password or any other parameter supporting Open Client 12.5 directory services, you must meet certain requirements for using directory services in your InfoMaker application. For details, see “Requirements for using Open Client directory services” in [Connecting to Your Database](#).

Corresponding CT-Lib connection property Specifying a value for DS_Password sets the corresponding SAP Sybase CT-Lib connection property named CS_DS_PASSWORD.

Examples To specify MYPASS as your application's password, type `MYPASS` in the Password box on the Directory Services page in the Database Profile Setup dialog box.

See also DS_Alias
DS_Copy
DS_DitBase
DS_Failover
DS_Principal
DS_Provider
DS_TimeLimit
Release

DS_Principal

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, DS_Principal is one of several parameters that you can set to enable network-based directory services in your application. (For other directory services parameters, see the See Also section.)

Some directory service providers and drivers require an authenticated principal (user ID) name to control an application's access to directory entries. For those providers and drivers, DS_Principal and DS_Password specify the principal name and password your application should use to identify you to the directory service provider.

You must specify a value for DS_Principal *before* connecting to the database in InfoMaker.

Using third-party directory service providers

For information about the third-party directory service providers and operating system platforms that Apeon has tested with Open Client directory services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

DS_Principal='principal_name'

Parameter	Description
<i>principal_name</i>	The principal (user ID) name your application should use to identify you to the directory service provider.

Default

None

InfoMaker does not set DS_Principal or the corresponding SAP Sybase Open Client Client-Library (CT-Lib) connection parameter CS_DS_PRINCIPAL if you do not specify a value.

Usage

When to use If your directory service provider requires an authenticated principal name for directory access, set DS_Principal to the principal (user ID) name that goes with your directory service password.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the [Release](#) parameter for more information.

Requirements for use To use DS_Principal or any other parameter supporting Open Client directory services, you must meet certain requirements for using directory services in your InfoMaker application. For details, see “Requirements for using Open Client directory services” in *Connecting to Your Database*.

Corresponding CT-Lib connection property Specifying a value for DS_Principal sets the corresponding SAP Sybase CT-Lib connection property named CS_DS_PRINCIPAL.

Examples

To specify JSMITH as your application's principal name, type **JSMITH** in the Principal Name box on the Directory Services page in the Database Profile Setup dialog box.

See also

DS_Alias
DS_Copy
DS_DitBase
DS_Failover
DS_Password
DS_Provider
DS_TimeLimit
Release

DS_Provider

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, DS_Provider is one of several parameters that you can set to enable network-based directory services in your application. (For other directory services parameters, see the See Also section.)

When you use Open Client directory services, you must specify your directory service provider names in the Open Client/Open Server Configuration utility so that the required drivers can be loaded for each provider. The default directory service provider is the one currently specified as active in the Configuration utility.

DS_Provider lets you specify a directory service provider name listed in the Open Client/Open Server Configuration utility *other than* the default (active) provider. The default value for DS_Provider is the provider name currently specified as active in the Configuration utility.

You must specify a value for DS_Provider *before* connecting to the database in InfoMaker.

Using third-party directory service providers

For information about the third-party directory service providers and operating system platforms that Apeon has tested with Open Client directory services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

DS_Provider='provider_name'

Parameter	Description
<i>provider_name</i>	The directory service provider name you want to use for directory services. The provider name is case sensitive. You must specify it <i>exactly as it appears</i> in the Open Client/Open Server Configuration utility.

Default

The default value for DS_Provider is the provider name currently specified as active in the Open Client/Open Server Configuration utility.

Usage

When to use Set DS_Provider to use a directory service provider specified in the Open Client/Open Server Configuration utility *other than* the default (active) provider. For instructions on using the Open Client/Open Server Configuration utility, see your SAP Sybase Open Client/Server configuration guide.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the [Release](#) parameter for more information.

Requirements for use To use DS_Provider or any other parameter supporting Open Client directory services, you must meet certain requirements for using directory services in your InfoMaker application. For details, see “Requirements for using Open Client directory services” in [Connecting to Your Database](#).

Corresponding CT-Lib connection property Specifying a value for DS_Provider sets the corresponding SAP Sybase CT-Lib connection property named CS_DS_PROVIDER.

Examples

To specify NTREGISTRY as the directory service provider name, type `NTREGISTRY` in the Provider box on the Directory Services page in the Database Profile Setup dialog box.

See also

[DS_Alias](#)
[DS_Copy](#)

DS_DitBase
DS_Failover
DS_Password
DS_Principal
DS_TimeLimit
Release

DS_TimeLimit

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, DS_TimeLimit is one of several parameters that you can set to enable network-based directory services in your application. (For other directory services parameters, see the See Also section.)

Some directory service providers and drivers support the use of time limits for a directory search. For those providers and drivers, DS_TimeLimit specifies the maximum number of seconds that a directory search lasts.

By default, DS_TimeLimit specifies that there is no time limit for a directory search.

You must specify a value for DS_TimeLimit *before* connecting to the database in InfoMaker.

Using third-party directory service providers

For information about the third-party directory service providers and operating system platforms that Apeon has tested with Open Client directory services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

DS_TimeLimit=*value*'

Parameter	Description
<i>value</i>	Specifies the maximum number of seconds that you want a directory search to last. You can also specify 'no_limit' (the default) to indicate that there is no time limit for the directory search. If the specified time limit expires and the target has not been found, the directory search is unsuccessful and the InfoMaker connection fails.

Default	DS_TimeLimit='no_limit'
Usage	<p><i>Set Release parameter</i> For DS_TimeLimit to take effect, you <i>must</i> also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the <i>Release</i> parameter for more information.</p> <p><i>Requirements for use</i> To use DS_TimeLimit or any other parameter supporting Open Client directory services, you must meet certain requirements for using directory services in your InfoMaker application. For details, see “Requirements for using Open Client directory services” in <i>Connecting to Your Database</i>.</p> <p><i>Corresponding CT-Lib connection property</i> Specifying a value for DS_TimeLimit sets the corresponding SAP Sybase CT-Lib connection property named CS_DS_TIMELIMIT.</p>
Examples	To specify that you want the directory search to last a maximum of 120 seconds (2 minutes), type <i>120</i> in the Directory Search Time Limit box on the Directory Services page in the Database Profile Setup dialog box.
See also	<i>DS_Alias</i> <i>DS_Copy</i> <i>DS_DitBase</i> <i>DS_Failover</i> <i>DS_Password</i> <i>DS_Principal</i> <i>DS_Provider</i> <i>Release</i>

Encrypt

Description Specifies that data should be encrypted before sending it over the network.

When to specify Encrypt

You must specify the Encrypt parameter *before* connecting to the database.

Controls SNC SQL Native Client for Microsoft SQL Server

Syntax Encrypt=*value*

Parameter	Description
<i>value</i>	Specifies whether data should be encrypted. This parameter is used in conjunction with the TrustServerCertificate DbParameter. Values are: <ul style="list-style-type: none">• 0 (Default) No encryption occurs.• 1 If TrustServerCertificate is not set, encryption occurs only if there is a verifiable server certificate. If TrustServerCertificate is set, encryption always occurs, but may use a self-signed server certificate.

Default

Encrypt=0

Usage

SQL Server 2005 always encrypts network packets associated with logging in to the server. If no certificate is provided on the server when it starts up, SQL Server generates a self-signed certificate that is used to encrypt login packets.

SQL Server Configuration Manager can be used to configure the SQL Native Client to request an encrypted connection using the Secure Sockets Layer (SSL), and to accept a self-signed certificate without validation. You can also request encryption by setting the Encrypt DbParameter to 1, which sets the SQL Native Client connection string keyword Encrypt. To enable encryption to be used when a certificate has not been provided on the server, set both Encrypt and TrustServerCertificate. The value of TrustServerCertificate is ignored if Encrypt is not set.

Examples

To specify that InfoMaker should encrypt data:

- **Database profile** Select the Encrypt Data check box on the System page in the Database Profile Setup dialog box.
- **Application** Type the following in code:

```
SQLCA.DbParameter="Encrypt=1"
```

See also

TrustServerCertificate

EncryptionPass

Description

Specifies a password for encrypting and decrypting data.

When to specify EncryptionPass

You must specify the EncryptionPass parameter *before* connecting to the database.

Controls I10 Informix

Syntax EncryptionPass='value'

Parameter	Description
value	A string that will be used as the password for encrypting and decrypting data.

Default None.

Usage In IDS 10.0 and later, the **SQL** statement **SET ENCRYPTION PASSWORD** can improve the confidentiality of data and support data integrity by defining or resetting a password for encryption and decryption of data at the column level.

You can set the EncryptionPass and Hint static DbParameters on the System tab page in the Database Profile Setup dialog box for I10 connections to specify a password and an optional hint to help you remember the password. The password does not display in the database Profile Setup dialog box and is encrypted in the database profile in the registry. The application uses built-in Informix functions to encrypt and decrypt character data.

Examples To specify Archimedes as the password for data encryption and Eureka as the hint to help you remember the password:

- **Database profile** Enter Archimedes in the Encryption Password box and Eureka in the Hint box on the System page in the Database Profile Setup dialog box.
- **Application** Type the following in code:

```
SQLCA.DbParameter="EncryptionPass='Archimedes',
Hint='Eureka' "
```

To specify Archimedes as the password for data encryption and Eureka as the hint to help you remember the password, enter Archimedes in the Encryption Password box and Eureka in the Hint box on the System page in the Database Profile Setup dialog box.

See also Hint

EncryptPassword

Description Specifies whether you want InfoMaker to encrypt your password automatically when connecting to an OLE DB data provider.

When to specify EncryptPassword

You must specify the EncryptPassword parameter *before* connecting to the database.

Controls

OLE DB

Syntax

EncryptPassword='value'

Parameter	Description
<i>value</i>	Specifies whether you want InfoMaker to encrypt your password. Values are: <ul style="list-style-type: none">• True Tells InfoMaker to encrypt the password• False (Default) Tells InfoMaker not to encrypt the password

Default

EncryptPassword='False'

Examples

To tell InfoMaker to encrypt your password when connecting to Microsoft SQL Server or an OLE DB data provider, select the Encrypt Password check box on the Security page in the Database Profile Setup dialog box.

See also

[DataLink](#)
[IntegratedSecurity](#)
[MaskPassword](#)
[PersistEncrypted](#)

FailoverPartner

Description

Specifies the name of a mirror server, allowing you to maintain database availability if a failover event occurs.

Controls

SNC SQL Native Client for Microsoft SQL Server

Syntax

FailoverPartner =value

Parameter	Description
<i>value</i>	Specifies the name of a mirror server.

Default

None

Usage

When failover occurs, the existing InfoMaker connection to SQL Server is lost. The SNC driver releases the existing connection and tries to reopen it. If reconnection succeeds, InfoMaker triggers the DBNotification event.

The following conditions must be satisfied for InfoMaker to trigger the failover event:

- The FailoverPartner DBParm is supplied at connect time
- The SQL Server database is configured for mirroring
- InfoMaker is able to reconnect successfully when the existing connection is lost

When failover occurs:

- InfoMaker returns an error code (998) and triggers the DBNotification event with notification type DBFailover!
- Existing cursors cannot be used and should be closed
- Any failed database operation can be tried again
- Any uncommitted transaction is lost. New transactions must be started

Examples

To set the name of a mirror server to use in the event of a failover, type the name of the mirror server in the Failover Partner text box on the System page of the Database Profile Setup dialog box.

FoDelay

Description

Specifies the amount of time (in milliseconds) you want InfoMaker to wait between attempts to fail over to another database server if the current database server goes down.

When to specify FoDelay

You must specify the FoDelay parameter *before* connecting to the database.

Controls

O90 Oracle9*i*
O10 Oracle 10*g*
ORA Oracle 11*g*/12*c*

Syntax

FoDelay='*value*'

Parameter	Description
<i>value</i>	Specifies the amount of time in milliseconds you want InfoMaker to wait between attempts to fail over to an another database server.

Default

FoDelay='10'

- Usage** You can enter a failover delay value only if you have enabled failover. This parameter cannot be set dynamically. The value set when the connection is made remains in effect until it is disconnected.
- Examples** To tell InfoMaker to wait 20 milliseconds between attempts to fail over, type **20** in the Delay box on the Network page in the Database Profile Setup dialog box.
- See also** [SvrFailover](#)

FoDialog

Description Specifies whether InfoMaker displays a runtime dialog box indicating when a failover occurs.

When to specify FoDialog

You must specify the FoDialog parameter *before* connecting to the database.

Controls ASE, SYC SAP Adaptive Server Enterprise
O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12c

Syntax FoDialog='value'

Parameter	Description
<i>value</i>	Specifies whether you want InfoMaker to display a runtime dialog box indicating when a failover occurs. Values are: <ul style="list-style-type: none">• No (Default) InfoMaker should not display a dialog box.• Yes InfoMaker should display a dialog box.

Default FoDialog='No'

Usage You can display a runtime dialog box only if you have enabled failover. This parameter cannot be set dynamically. The value set when the connection is made remains in effect until it is disconnected.

Examples To tell InfoMaker to display a runtime dialog box when a failover occurs, select the Display Runtime Dialog When Failing Over check box on the Network page in the Database Profile Setup dialog box.

See also [SvrFailover](#)

FoRetryCount

Description Specifies the number of times you want InfoMaker to try to fail over to another database server if the current database server goes down.

When to specify FoRetryCount

You must specify the FoRetryCount parameter *before* connecting to the database.

Controls O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12/c

Syntax FoRetryCount='value'

Parameter	Description
value	Specifies the number of times you want InfoMaker to try to fail over.

Default FoRetryCount='10'

Usage You can enter a failover retry value only if you have enabled failover.

This parameter cannot be set dynamically. The value set when the connection is made remains in effect until it is disconnected.

Examples To tell InfoMaker to try 20 times to fail over, enter the value 20 in the Retry Count box on the Network page in the Database Profile Setup dialog box.

See also SvrFailover

FormatArgsAsExp

Description Controls whether InfoMaker converts a report retrieval argument of decimal datatype to scientific (exponential) notation if the argument exceeds 12 digits but has fewer than 16 digits. If FormatArgsAsExp is set to Yes (the default), InfoMaker performs this conversion.

When to specify FormatArgsAsExp

You must specify a value for FormatArgsAsExp *before* connecting to the database.

Controls ASE, SYC SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect

JDB JDBC
 ODBC interface
 O90 Oracle9i
 O10 Oracle 10g
 ORA Oracle 11g/12c

Syntax

FormatArgsAsExp= '*value*'

Parameter	Description
<i>value</i>	<p>Specifies whether you want InfoMaker to convert a report retrieval argument of decimal datatype to scientific (exponential) notation if the argument exceeds 12 digits but has fewer than 16 digits. Values are:</p> <ul style="list-style-type: none"> • Yes InfoMaker converts a retrieval argument of decimal datatype to scientific notation if it exceeds 12 digits but has fewer than 16 digits. • No (Default) InfoMaker leaves the retrieval argument as a decimal and does not perform the default conversion to scientific notation if it exceeds 12 digits but has fewer than 16 digits.

Default

FormatArgsAsExp='No'

Usage

When to use The setting of FormatArgsAsExp might affect the speed of data retrieval in your reports, especially if you are accessing large databases.

If FormatArgsAsExp is set to Yes, InfoMaker converts retrieval arguments of type decimal to scientific notation if the argument exceeds 12 digits but has fewer than 16 digits. Some DBMS optimizers might interpret the resulting scientific notation as a different datatype and scan all rows in the table to find it. This can slow data retrieval if, for example, you are accessing a DB2 database with many large tables.

Setting FormatArgsAsExp to No tells InfoMaker to leave the retrieval argument as a decimal and not convert it to scientific notation. This speeds data retrieval for large databases.

Retrieval argument size limited

The FormatArgsAsExp parameter is relevant only if a retrieval argument of type decimal has fewer than 16 digits.

Examples

To tell InfoMaker to convert a retrieval argument exceeding 12 digits but with fewer than 16 digits to scientific notation, check the Format Arguments in Scientific Notation check box on the Syntax page (or Transaction page in the case of the DIR interface) in the Database Profile Setup dialog box.

GenerateEqualsNull

Description Specifies how DataWindows generates `=null` and `<> null` expressions in retrieval arguments.

Controls All database interfaces

Syntax `GenerateEqualsNull= 'value'`

Parameter	Description
<i>value</i>	<ul style="list-style-type: none"> • Yes Does not change <code>=null</code> or <code><> null</code> expressions. • No Converts <code>=null</code> expression to <code>IS NULL</code>, <code><> null</code> to <code>IS NOT NULL</code>.

Default `GenerateEqualsNull='No'`

Usage *When to use* When a DataWindow retrieves data from tables that contain null columns, most DBMS interfaces expect expressions of the form, `IS NULL` and `IS NOT NULL`. Expressions that reference null data as values, such as `Where column = NULL`, can cause the DBMS to reject the retrieval. For this reason, such expressions in DataWindows are normally converted to the standard ANSI syntax during retrieval.

If your DataWindow retrieves null data from a DBMS that supports expressions of the `=null` or `<> null` form, and you want to suppress the conversion of those expressions to standard syntax, you can set the `GenerateEqualsNull` DBParm to true.

Examples Consider these two statements:

```
SELECT "a1"."c1" FROM "a1" WHERE "a1"."c2" = :p1
SELECT "a1"."c1" FROM "a1" WHERE "a1"."c2" <> :p1
```

If `GenerateEqualsNull` is set to false, the statements are generated as:

```
SELECT "a1"."c1" FROM "a1" WHERE "a1"."c2" is null
SELECT "a1"."c1" FROM "a1" WHERE "a1"."c2" is not null
```

If `GenerateEqualsNull` is set to true, the statements are generated as:

```
SELECT "a1"."c1" FROM "a1" WHERE "a1"."c2" = null
SELECT "a1"."c1" FROM "a1" WHERE "a1"."c2" <> null
```

HANotification

Description Specifies whether a High Availability (HA) client connected to an RAC database will be notified if the database server shuts down.

Controls O10 Oracle 10g

Syntax

HANotification=*value*

Parameter	Description
<i>value</i>	A value specifying whether the client will be notified if the database server shuts down. Values are: <ul style="list-style-type: none">• No (Default) Do not notify the client if the server shuts down.• Yes Notify the client if the server shuts down.

Default

HANotification="No"

Usage

Oracle Real Application Clusters (RAC) is a cluster database that uses a shared cache architecture. In Oracle 10g Release 2, a High Availability (HA) client connected to an RAC database can register a callback to indicate that it wants the server to notify it in case of a database failure event that affects a connection made by the client.

Examples

To specify that the HA client should be notified when the database server shuts down, select the HANotification check box on the System page in the Database Profile Setup dialog box.

Hint

Description

Specifies a hint to assist in remembering the password specified for encrypting and decrypting data.

When to specify Hint

You must specify the Hint parameter *before* connecting to the database.

Controls

I10 Informix

Syntax

Hint='*value*'

Parameter	Description
<i>value</i>	A string that will be used as a hint to help you remember the password used for encrypting and decrypting data.

Default

None.

Usage

In IDS 10.0 and later, the SQL statement **SET ENCRYPTION PASSWORD** can improve the confidentiality of data and support data integrity by defining or resetting a password for encryption and decryption of data at the column level.

You can set the EncryptionPass and Hint static DbParameters on the System tab page in the Database Profile Setup dialog box for I10 connections to specify a password and an optional hint to help you remember the password. The password does not display in the database Profile Setup dialog box and is encrypted in the database profile in the registry. The application uses built-in Informix functions to encrypt and decrypt character data.

Examples

To specify Archimedes as the password for data encryption and Eureka as the hint to help you remember the password:

- **Database profile** Enter Archimedes in the Encryption Password box and Eureka in the Hint box on the System page in the Database Profile Setup dialog box.
- **Application** Type the following in code:

```
SQLCA.DbParameter="EncryptionPass='Archimedes',  
Hint='Eureka'"
```

To specify Archimedes as the password for data encryption and Eureka as the hint to help you remember the password, enter Archimedes in the Encryption Password box and Eureka in the Hint box on the System page in the Database Profile Setup dialog box.

See also

[EncryptionPass](#)

Host

Description

If your DBMS supports it, specifies the workstation name when connecting to the database in InfoMaker. The Host parameter lets you assign any 10-character label to identify the process you are about to create when you connect to the database. This label helps you distinguish your process from others running on the database server.

When to specify Host

You must specify the Host parameter *before* connecting to the database in InfoMaker.

Controls

ASE, SYC SAP Adaptive Server Enterprise
SNC SQL Native Client for Microsoft SQL Server

Syntax

Host='*workstation_name*'

Default

None

Usage	<p>For Adaptive Server, when you specify a value for Host, InfoMaker sets the CS_HOSTNAME connection property to the workstation name you specify.</p> <p>The value you specify for the Host parameter displays in the hostname column of the MASTER.DBO.SYSPROCESSES table in a SQL Server database. How you use the Host parameter depends on the design of your InfoMaker application.</p> <p>For example, many sites want to secure their production tables so that updates are possible only through a specific application. To do this, you can grant explicit authority to the InfoMaker application but <i>not</i> to users. The application prompts the user for an authorization ID and password, verifies it, and then connects to the database through a single application login ID. Only this application login ID has authorization to update production tables.</p> <p>In this scenario, you can use the Host parameter to store the name of the user running the application.</p>
Examples	<p>Example 1 To set the host name to Alan, type <code>Alan</code> in the Workstation Name box on the Network page in the Database Profile Setup dialog box.</p> <p>Example 2 You can use the Host and AppName parameters together to specify both the host name and the application name. To set the host name to Jane and the application name to Sales, type Jane in the Workstation Name box and Sales in the Application Name box on the Network page in the Database Profile Setup dialog box.</p>
See also	AppName

HostReqOwner

Description	<p>Specifies the name of the host request library defined in a DB2/MVS database.</p> <hr/> <p>When to specify HostReqOwner You must specify the HostReqOwner parameter <i>before</i> connecting to the database.</p> <hr/>
Controls	<p>DIR SAP Sybase DirectConnect (applies only to Access Service for DB2/MVS and Open ServerConnect™)</p>
Syntax	<pre>HostReqOwner='owner_id'</pre>
Default	<pre>HostReqOwner='Appeon'</pre>

Usage The host request library is a special DB2 table that stores host-resident requests. A host-resident request is a **SQL** statement that a client application can execute as a procedure. If you do not use Apeon as the owner name for this host request library, you should set the HostReqOwner parameter to an appropriate name for your site.

TRS Support

The HostReqOwner parameter is not applicable to DirectConnect TRS connections.

Examples To set the name of your host request library to Stratus, type **Stratus** in the Host Request Lib Owner box on the System page in the Database Profile Setup dialog box.

See also [TRS](#)
[UseProcSyntax](#)

IdentifierQuoteChar

Description Specifies the single quote character you want InfoMaker to use to delimit the names of identifiers (tables, columns, indexes, and constraints) when it generates **SQL** statements. InfoMaker uses the quote character you specify instead of the default quote character returned by your driver or data provider.

DelimitIdentifier must be set to Yes

In order for IdentifierQuoteChar to take effect, the DelimitIdentifier parameter must be set to Yes. Otherwise, InfoMaker's default behavior is *not* to delimit identifiers in **SQL** statements and to ignore any value specified for IdentifierQuoteChar.

Controls JDB JDBC
ODBC (if driver and back-end DBMS support this feature)
OLE DB

Syntax IdentifierQuoteChar = '*quote_character*'

Parameter	Description
<i>quote_character</i>	The single character you want InfoMaker to use instead of your driver's or data provider's default quote character to delimit the names of identifiers in SQL statements.

Default None

InfoMaker searches the following in this order to determine the IdentifierQuoteChar value:

- 1 The section for your database profile in the InfoMaker initialization file
- 2 The section for your ODBC driver in the PBODB170 initialization file or the section for your JDBC driver in the registry

If InfoMaker does not find an IdentifierQuoteChar value in these locations, it makes a SQLGetInfo call to your driver to return the default SQL_IDENTIFIER_QUOTE_CHAR value.

When using the OLE DB interface

If no value is specified for the IdentifierQuoteChar parameter, InfoMaker does not use a quote character.

Usage

By default, some drivers return quote characters that do not work with InfoMaker's parsing routines, such as the backquote character (`). As a result, delimiting is turned off for these drivers in InfoMaker.

However, if you paint **SQL** statements containing identifiers that require delimiters, syntax errors can occur if you are using a driver for which delimiting is turned off. To avoid such errors, set IdentifierQuoteChar to override the driver's default quote character.

Examples

To specify *c* as the quote character you want InfoMaker to use to delimit identifiers in SQL statements, type *c* in the Identifier Quote Character box on the Syntax page in the Database Profile Setup dialog box.

See also

[DelimitIdentifier](#)

Identity

Description

Specifies the syntax the **SNC** database interface uses to obtain the identity value of a new row in a database table.

When a report update request inserts a new row into a Microsoft **SQL** Server table that contains an **IDENTITY** column, the report engine calls the **SNC** interface to obtain the identity value of the newly inserted row. The Identity database parameter allows you to define how this request is implemented.

Controls

SNC SQL Native Client for Microsoft **SQL** Server

Syntax

Identity=*value*

Parameter	Description
<i>value</i>	<p>A value specifying the syntax for obtaining the identity value of a newly inserted row. Values are:</p> <ul style="list-style-type: none"> • @@IDENTITY (Default) Use the syntax <code>SELECT @@identity</code>. • IDENT_CURRENT() Use the syntax <code>SELECT IDENT_CURRENT('tablename')</code>. • SCOPE_IDENTITY() Use the syntax <code>SELECT scope_identity()</code>.

Default

Identity="@@IDENTITY"

Usage

By default, the **SNC** interface issues `SELECT @@identity` to obtain the **IDENTITY** column value of the newly inserted row. It returns the last **IDENTITY** value produced on a connection, regardless of the table that produced the value, and regardless of the scope of the statement that produced the value.

`SELECT IDENT_CURRENT('tablename')` returns the last **IDENTITY** value produced in a table, regardless of the connection that created the value, and regardless of the scope of the statement that produced the value.

`SELECT SCOPE_IDENTITY()` returns the last **IDENTITY** value produced on a table and by a statement in the same scope, regardless of the table that produced the value.

Because Identity is a dynamic database parameter, it can be set and reset at any time during an application.

Examples

To specify the syntax for obtaining the identity value of a newly inserted row, select the Use `@@identity` to obtain **IDENTITY** Column Value check box on the Syntax page in the Database Profile Setup dialog box.

ImpersonationLevel

Description

Specifies the level of impersonation that the data server is allowed to use when impersonating its OLE DB data provider and InfoMaker. This parameter applies only to network connections other than Remote Procedure Call (RPC) connections.

When to specify ImpersonationLevel

You must specify the ImpersonationLevel parameter *before* connecting to the database.

Controls

OLE DB

Syntax

ImpersonationLevel=*value*'

Parameter	Description
<i>value</i>	<p>Specifies the level of impersonation. Values are:</p> <ul style="list-style-type: none">• Not set No level of impersonation is selected.• Anonymous The client is anonymous to the server and the server process cannot obtain identification information about the client and cannot impersonate the client.• Delegate The process can impersonate the client's security context while acting on behalf of the client. The server process can also make outgoing calls to other servers while acting on behalf of the client.• Identify The server can obtain the client's identity. The server can impersonate the client for ACL checking but cannot access system objects as the client.• Impersonate The server process can impersonate the client's security context while acting on behalf of the client. This information is obtained when the connection is established, not on every call.

Default

ImpersonationLevel='Not set'

Examples

To set a level of impersonation to anonymous, on the Security page in the Database Profile Setup dialog box, select Anonymous from the Impersonation Level drop-down list.

See also

[DataLink](#)

INET_DBPATH

Description Specifies the Informix DBPATH setting. The DBPATH environment variable identifies a list of directories that contain Informix databases. INET_DBPATH typically specifies the location of Informix databases if this is *other* than in a directory on the database server.

Controls I10 Informix
IN9 Informix

Syntax INET_DBPATH='*server_db_path*'

Parameter	Description
<i>server_db_path</i>	The name of the directory containing Informix databases

Default By default, InfoMaker uses the value specified for DBPATH in the *HKEY_LOCAL_MACHINE\SOFTWARE\Informix\Environment* registry key.

Examples **Example 1** To specify that the directory /HOME/Informix contains Informix databases, type */home/Informix* in the Database Path box on the Network page in the Database Profile Setup dialog box.

Example 2 You can specify values for INET_DBPATH, INET_PROTOCOL, and INET_SERVICE together. To specify that the directory */Informix* contains Informix databases, and that you want to connect using the turbo service and theolsoctcp network protocol:, type */Informix* in the Database Path box, turbo in the Service Name box, andolsoctcp in the Protocol Type box on the Network page in the Database Profile Setup dialog box.

See also INET_PROTOCOL
INET_SERVICE

INET_PROTOCOL

Description Specifies the network protocol that the Informix client software uses to communicate with a remote Informix database server.

Controls I10 Informix
IN9 Informix

Syntax INET_PROTOCOL='*network_protocol*'

Parameter	Description
<i>network_protocol</i>	A string that specifies the name of the network protocol used by the Informix client software. For information about the correct network protocol for your site, see your Informix system administrator.

Default

By default, InfoMaker uses the network protocol specified in the *HKEY_LOCAL_MACHINE\SOFTWARE\Informix\SqlHosts* registry key.

Examples

Example 1 To specify that Informix client software uses the Novell IPX/SPX network protocol, type *ipx* in the Protocol Type box on the Network page in the Database Profile Setup dialog box.

Example 2 You can specify values for INET_DBPATH, INET_PROTOCOL, and INET_SERVICE together. To specify that the directory */Informix* contains Informix databases, and that you want to connect using the turbo service and the olsoctcp network protocol, type */Informix* in the Database Path box, turbo in the Service Name box, and olsoctcp in the Protocol Type box on the Network page in the Database Profile Setup dialog box.

See also

INET_DBPATH
INET_SERVICE

INET_SERVICE

Description

Specifies the name of the service that a remote Informix database server uses to listen to all incoming requests from client applications.

Controls

I10 Informix
IN9 Informix

Syntax

INET_SERVICE='*service_name*'

Parameter	Description
<i>service_name</i>	A string that specifies the name of the service that a remote Informix database server uses to listen to incoming requests For information about the correct service name for your site, see your Informix system administrator.

Default

By default, InfoMaker uses the service name specified in the *HKEY_LOCAL_MACHINE\SOFTWARE\Informix\SqlHosts* registry key.

Examples

Example 1 To specify that your Informix database server uses the sqlexec service name, type `sqlexec` in the Service Name box on the Network page in the Database Profile Setup dialog box.

Example 2 You can specify values for `INET_DBPATH`, `INET_PROTOCOL`, and `INET_SERVICE` together. To specify that the directory `/Informix` contains Informix databases, and that you want to connect using the turbo service and the olsoctcp network protocol, type `/Informix` in the Database Path box, `turbo` in the Service Name box, and `olsoctcp` in the Protocol Type box on the Network page in the Database Profile Setup dialog box.

See also

`INET_DBPATH`
`INET_PROTOCOL`

Init_Prompt

Description

Specifies whether you want to be prompted during initialization.

When to specify Init_Prompt

You must specify the `Init_Prompt` parameter *before* connecting to the database.

Controls

OLE DB

Syntax

Prompt='value'

Parameter	Description
<i>value</i>	<p>Specifies whether you want to be prompted during initialization. Values are:</p> <ul style="list-style-type: none"> • Not set Do not prompt. • Always Always prompt for initialization information. • If needed Prompt only if more information is needed. • If needed (required) Prompt only if more information is needed. Do not allow the user to enter optional information. • Never Do not prompt.

Default

`Init_Prompt='Not set'`

Examples

To specify that you want always to be prompted during initialization:

- **Database profile** Select Always from the Prompt drop-down list on the System page in the Database Profile Setup dialog box.
- **Application** Type the following in code:

```
SQLCA.DbParameter="INIT_PROMPT='DBPROMPT_PROMPT'"
```

To specify that you want always to be prompted during initialization, select Always from the Prompt drop-down list on the System page in the Database Profile Setup dialog box.

See also

[DataLink](#)

InsertBlock

Description

Specifies the number of rows that you want the Data Pipeline in InfoMaker to insert at one time into a table in the destination database.

For instructions on using the Data Pipeline, see the *Users Guide*.

Controls

ODBC (only in Data Pipeline if driver and back-end DBMS support this feature)

Syntax

InsertBlock=*insert_blocking_factor*

Parameter	Description
<i>insert_blocking_factor</i>	The number of rows that you want the Data Pipeline to insert at one time into a table in the destination database, up to a maximum of 100 rows (Default=100 rows). To turn off block inserting for an ODBC data source in the Data Pipeline, set InsertBlock to 1 or DisableBind to 1 in the database profile of the destination database.

Default

InsertBlock=100

Usage

Requirements for using InsertBlock To use the InsertBlock parameter, *all* of the following must be true:

- You are using an ODBC driver to access the destination database in the Data Pipeline.
- The destination database supports the use of bind variables. (For more about bind variables, see [DisableBind](#).)
- The DisableBind parameter is not set to 1 (the default is 0) in the database profile of the destination database. This enables the default binding of input parameters to a compiled SQL statement in InfoMaker.
- Maximum Errors is set to 1 in the Data Pipeline.

The SQL Anywhere ODBC driver meets the first two requirements.

To determine whether your ODBC driver meets these requirements, see the documentation that comes with your driver.

Determining the InsertBlock value InfoMaker searches the following in this sequence to determine the value for InsertBlock:

- 1 The section for your database profile in the InfoMaker initialization file
- 2 The section for your ODBC driver in the PBODB170 initialization file

If InfoMaker does not find an InsertBlock value in these locations, it defaults to an insert blocking factor of 100 rows.

What happens When InfoMaker finds a value for InsertBlock, the Data Pipeline batches the specified number of rows and inserts them with a single call to the ODBC driver you are using to access the destination database.

If you specify an InsertBlock value or Data Pipeline commit factor of fewer than 100 rows, the Data Pipeline batches and inserts the specified number of rows into the destination database. If you specify more than 100 rows, the Data Pipeline batches and inserts at most only 100 rows at one time.

The insert blocking factor that the Data Pipeline actually uses depends on the size of the data in each column inserted in the destination database. In addition, the Data Pipeline does not exceed 64K of data in the buffer for any one column.

Turning off block inserting To turn off block inserting for an ODBC data source in the Data Pipeline, you can do any of the following in the database profile of the destination database:

- Set the InsertBlock parameter to 1
- Set the DisableBind parameter to 1 (to disable default binding of input parameters to a compiled SQL statement)
- In the Data Pipeline, set Maximum Errors to a value other than 1

Examples

To set the insert blocking factor in the Data Pipeline to 50 rows:

- **Database profile** Type 50 in the Insert Blocking Factor box on the Transaction page in the Database Profile Setup dialog box.
- **Application** Type the following in code:

```
SQLCA.DbParameter="InsertBlock=50"
```

To set the insert blocking factor in the Data Pipeline to 50 rows, type 50 in the Insert Blocking Factor box on the Transaction page in the Database Profile Setup dialog box.

See also

[DisableBind](#)

IntegratedSecurity

Description

Specifies the name of the authentication service used by the data server to identify the user.

If this parameter is specified, none of the other OLE DB authentication parameters (CacheAuthentication, EncryptPassword, MaskPassword, PersistEncrypted, and PersistSecurityInfo) are needed and are ignored if specified.

When to specify IntegratedSecurity

You must specify the IntegratedSecurity parameter *before* connecting to the database.

Controls

OLE DB

Syntax

IntegratedSecurity=*value*

Parameter	Description
<i>value</i>	A string specifying the name of the authentication service. If NULL, the default authentication service is used.

Default

None

Examples

To use an authentication service such as the Security Support Provider Interface (SSPI) for Windows NT, type the name of the authentication service in the Integrated Security box on the Security page in the Database Profile Setup dialog box.

KeepAlive

Description

Determines whether packets are sent to the database to ensure that the connection is still active.

When to specify KeepAlive

You must specify the KeepAlive parameter *before* connecting to the database.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

KeepAlive=*value*

Parameter	Description
<i>value</i>	Specifies whether the Open Client/Server CS_CON_KEEPALIVE property is set for your connection. Values are: <ul style="list-style-type: none"> • 1 (Default) CS_CON_KEEPALIVE property is set. • 0 CS_CON_KEEPALIVE property is not set.

Default

KeepAlive=1

Usage

KeepAlive sets the value of the SAP Sybase CT-Lib connection property CS_CON_KEEPALIVE to **true** or **false**. The default setting ensures that your connection is alive by sending packets to the database when the connection is idle. Set the value of this property to false for mobile clients that do not maintain constant connections.

Examples

To set the KeepAlive value to 0 when you do not want to maintain a connection, clear the Keep Connection Alive check box on the Network page.

Language

Description

For those interfaces that support it, specifies the language you want to use when connecting to your target database.

When to specify Language

You must specify the Language parameter *before* connecting to the database. The Language setting takes effect when you connect to the database.

Controls

ASE, SYC SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect

Syntax

Language='language_name'

Default

None

Usage

When you specify a value for Language, InfoMaker:

- Allocates a CS_LOCALE structure for this connection
- Sets the CS_SYB_LANG value to the language you specify
- Sets the **SQL** Server CS_LOC_PROP connection property with the new locale information

If you have previously set a value for the Locale parameter, which includes settings for the language and character set you want the Open Client software to use, you can override the language value by specifying a new value for the Language parameter and reconnecting to the database.

Unicode data access InfoMaker can access Unicode data in an Adaptive Server Enterprise (ASE) 12.5 or higher Unicode database or in Unicode columns in ASE 12.5 or higher. InfoMaker converts between double-byte character set (DBCS) data and Unicode automatically, provided that the Language and CharSet parameters are set with DBCS values (or the Locale parameter is set with DBCS values).

For example:

```
Language='tchinese'  
CharSet='big5'
```

Examples

To set the Language parameter to French, type [French](#) in the Language box on the Connection page or Regional Settings page in the Database Profile Setup dialog box.

See also

[CharSet](#)
[Locale](#)

LCID

Description

Specifies the locale identifier that you want the OLE DB data provider to use.

When to specify LCID

You must specify the LCID parameter *before* connecting to the database.

Controls

OLE DB

Syntax

```
LCID='lcid_name'
```

Default

None

Usage

You specify the locale identifier at initialization. This provides a way for the data server to determine InfoMaker's preferred locale language and character set. However, setting this parameter does not guarantee that all text returned to InfoMaker is translated according to the locale ID.

Examples

To set the locale to US English, type [1033](#) in the LCID box on the System page in the Database Profile Setup dialog box.

See also

[CharSet](#)

Language

Locale**Description**

Specifies the locale name that you want the SAP Sybase Open Client software to use when connecting to an SAP Adaptive Server Enterprise database or a database accessed through DirectConnect in InfoMaker.

When to specify Locale

You must specify the Locale parameter *before* connecting to the database.

Controls

ASE, SYC SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect

Syntax

Locale='locale_name'

Default

The default locale defined in your *LOCALES.DAT* file

Usage

Locales Locales are stored as entries in a file named *LOCALES.DAT*. The *LOCALES.DAT* file contains information about the languages and character sets you are using with the SAP Sybase Open Client software. The SAP Sybase Open Client installation places the *LOCALES.DAT* file in the *\$\$SYBASE\LOCALES* directory.

An entry in the *LOCALES.DAT* file has the following format:

```
locale=locale_name, language_name, character_set_name
```

For example:

```
locale=default, us_english, cp850
locale=enu, us_english, cp850
locale=fra, french, cp850
```

Why set Locale parameter Setting a value for the Locale parameter lets you use a locale *other than the default locale* when accessing an Adaptive Server Enterprise or DirectConnect database. If you do not set a value for Locale, SAP Sybase Open Client uses the default locale defined in your *LOCALES.DAT* file.

What happens When you specify a value for the Locale parameter, InfoMaker:

- Allocates a CS_LOCALE structure for this connection
- Sets the CS_LC_ALL value to the locale name you specify

- Sets the SQL Server CS_LOC_PROP connection property with the new locale information

Overriding Locale parameter If you have previously set a value for the Locale parameter that includes settings for the language and character set you want to use, you can override the language or character set values by specifying new values for the Language or CharSet parameter and reconnecting to the database.

Unicode data access The DataWindow server can access Unicode data in an ASE 12.5 or later Unicode database or in Unicode columns in ASE 12.5 or later. The DataWindow server converts between double-byte character set (DBCS) data and Unicode automatically, provided that the Locale parameter is set with DBCS values. For example, the Locale parameter should be set to `chs` or `cht`.

Examples

To set the locale to `fra`, type `fra` in the Locale box on the Regional Settings page in the Database Profile Setup dialog box. Setting the Locale parameter to `fra` has the same effect as individually setting both the Language and CharSet parameters as follows:

```
Language='French'  
CharSet='cp850'
```

See also

[CharSet](#)
[Language](#)

Location

Description

Specifies the location of the data source to which you want your OLE DB data provider to connect. Typically the location is the database server name.

When to specify Location

You must specify the Location parameter *before* connecting to the database.

Controls

OLE DB

Syntax

Location='location_name'

Default

None

Usage

Implementation of the Location parameter varies depending on the OLE DB data provider you are using. For specific information, see the data provider documentation provided by the OLE DB vendor.

Log

Description Specifies whether the database server should log updates of text and image data in the transaction log. By default, the database server logs updates of text and image data in the transaction log.

Controls ASE, SYC SAP Adaptive Server Enterprise

Syntax Log=*value*

Parameter	Description
<i>value</i>	<p>A value that specifies whether the database server should log updates of text and image data in the transaction log. Values are:</p> <ul style="list-style-type: none"> • 0 Do not log text and image updates in the transaction log. Specify this value only if your database server allows you to disable logging. • 1 (Default) Log text and image updates in the transaction log.

Default Log=1

Usage You should set the Log parameter to 0 only if your database server allows you to disable logging.

Examples To specify that InfoMaker should *not* log text and image updates in the transaction log, clear the Log Text and Image Updates check box on the System page or Transaction page in the Database Profile Setup dialog box.

LoginTimeOut

Description Specifies the number of seconds the JDBC or ODBC driver should wait for a login request to a JDBC database or an ODBC data source.

Controls ODBC (if driver and back-end DBMS support this feature)

Syntax LoginTimeOut=*value*

Parameter	Description
<i>value</i>	The number of seconds you want the driver to wait for a login request

Default ODBC: LoginTimeOut=15; JDBC: LoginTimeOut=0

Usage If you set LoginTimeOut to 0, InfoMaker does not call the JDBC or ODBC driver to set the LoginTimeOut value and instead waits the number of seconds specified by the JDBC or ODBC driver's client software. If you set LoginTimeOut to a value greater than 0, InfoMaker does call the JDBC or ODBC driver to set the LoginTimeOut value.

Examples To set the LoginTimeOut value to wait 60 seconds for a login request, type 60 in the Login Timeout box on the Network page in the Database Profile Setup dialog box.

LowerCaseIdent

Description Specifies whether InfoMaker displays identifier names in lowercase.

Controls DIR SAP Sybase DirectConnect (applies only to DB2/MVS)

Syntax LowerCaseIdent='value'

Parameter	Description
value	Specifies whether you want InfoMaker to display identifier names in lowercase. Values are: <ul style="list-style-type: none">• Yes Display identifier names in lowercase• No (Default) Do not display identifier names in lowercase

Default LowerCaseIdent='No'

Usage InfoMaker displays identifier names in uppercase (the way they are stored in the database). The LowerCaseIdent parameter can be set only if the DelimitIdentifier parameter is set to No, indicating that InfoMaker should not enclose table and column names in double quotes. If you try to enclose a table and column names in double quotes with identifier names in lowercase, the LowerCaseIdent parameter value is reset to the default value, and you receive a warning message.

Migrating PBMDI and PBNET applications to PBDIR

If you are migrating an application that previously used the InformationConnect DB2 Gateway or Net-Gateway for DB2 interface to the DirectConnect for DB2/MVS database interface, you should set the LowerCaseIdent parameter value to Yes. This enables you to continue to use the Select painter to edit reports.

Examples To have InfoMaker display identifier names in lowercase, select the Display Identifiers In Lower Case check box on the Syntax page in the Database Profile Setup dialog box.

MapDateToDateTime

Description Maps the Oracle `Date` datatype to the `DateTime` datatype to enable the time part of the data to be retrieved.

Controls JDB JDBC

Syntax `MapDateToDateTime=value`

Parameter	Description
<i>value</i>	<p>Specifies whether the Oracle <code>Date</code> datatype is mapped to the <code>DateTime</code> datatype. Values are:</p> <ul style="list-style-type: none"> • 0 (Default) The Oracle <code>Date</code> datatype is not mapped to the <code>DateTime</code> datatype. The time part of the <code>Date</code> column is not retrieved. • 1 The Oracle <code>Date</code> datatype is mapped to the <code>DateTime</code> datatype. The time part of the <code>Date</code> column is retrieved.

Default `MapDateToDateTime=0`

Usage The Oracle `Date` datatype stores both date and time data. When you connect to Oracle using the Oracle JDBC Thin Driver, only the date part of the data is retrieved. To retrieve the time part of the data, set the `MapDateToDateTime` database parameter to map the datatype of Oracle `Date` columns to the `DateTime` datatype.

Examples To retrieve both date and time data from an Oracle `Date` column:

- **Database profile** Select the Map Date to DateTime check box on the Transaction page in the Database Profile Setup dialog box.

To retrieve both date and time data from an Oracle `Date` column, select the Map Date to DateTime check box on the Transaction page in the Database Profile Setup dialog box.

MaskPassword

Description Specifies whether you want InfoMaker to mask your password automatically when connecting to an OLE DB data provider.

When to specify MaskPassword

You must specify the `MaskPassword` parameter *before* connecting to the database.

Controls OLE DB

Syntax

MaskPassword=*value*

Parameter	Description
<i>value</i>	Specifies whether you want InfoMaker to mask your password. Values are: <ul style="list-style-type: none">• True Tells InfoMaker to mask the password• False (Default) Tells InfoMaker not to mask the password

Default

MaskPassword='False'

Examples

To tell InfoMaker to mask your password when connecting to an OLE DB data provider, select the Mask Password check box on the Security page in the Database Profile Setup dialog box.

See also

DataLink
EncryptPassword
IntegratedSecurity
PersistEncrypted

MaxConnect

Description

Sets the maximum number of simultaneous connections you want to make when accessing a database.

The default is 25 simultaneous connections. You can override this default by setting MaxConnect up to the maximum number of simultaneous connections configured on the database server.

When to specify MaxConnect

You must specify a value for the MaxConnect parameter *before* connecting to the database.

Controls

ASE, SYC SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect

Syntax

MaxConnect=*value*

Parameter	Description
<i>value</i>	The maximum number of simultaneous connections you want to make when accessing a database

Default

MaxConnect=25

Usage *DirectConnect, ASE, and SYC* MaxConnect sets the SAP Sybase CT-Lib connection property CS_MAX_CONNECT to the number of simultaneous database connections you specify for a single CT-Lib context.

Examples To set the MaxConnect value to a maximum of 50 simultaneous database connections, type 50 in the Maximum Client Library Connections box (when using the ASE or SYC interface) or the Maximum Connections For This Context box (when using the DIR interface). This check box is on the Network page.

MixedCase

Description Specifies whether you want connections to an Oracle database to be case sensitive or case insensitive.

By default, MixedCase is set to 0. This setting specifies a case-insensitive connection and assumes that all identifiers are uppercase. To make the Oracle connection case sensitive, set the MixedCase parameter to 1.

Controls O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12c

Syntax MixedCase=*value*

Parameter	Description
<i>value</i>	Specifies whether an Oracle database connection is case sensitive or case insensitive. Values are: <ul style="list-style-type: none"> • 0 (Default) The Oracle database connection is case insensitive. It assumes that all identifiers are uppercase. • 1 The Oracle database connection is case sensitive. It supports mixed case, uppercase, and lowercase identifiers.

Default MixedCase=0

Usage When you set the MixedCase parameter to 1 and define a primary key for a table in an Oracle database, all of the following must contain only uppercase letters:

- The name of the primary key
- The name of the table containing the primary key
- The names of any foreign keys that reference the primary key

Examples

To make an Oracle database connection case sensitive, select the Case Sensitive check box on the Connection page in the Database Profile Setup dialog box.

Mode

Description

Specifies access permission to the OLE DB data provider.

When to specify Mode

You must specify the Mode parameter *before* connecting to the database.

Controls

OLE DB

Syntax

Mode='value'

Parameter	Description
<i>value</i>	Specifies access permission to the OLE DB data provider. Values are: <ul style="list-style-type: none">• Deny read share Prevents other users from opening in read mode.• Deny write share Prevents other users from opening in write mode.• Exclusive share Prevents other users from opening in read/write mode.• No share deny Neither read nor write access can be denied to other users.• Read/Write Allows read/write access.• Read-only Allows read access.• Write-only Allows write access.

Default

None

Examples

To allow other users read/write access to the OLE DB data provider, on the Transaction page in the Database Profile Setup dialog box, select Read/Write from the Mode list box.

See also

[DataLink](#)

MsgTerse

Description Specifies whether InfoMaker should display terse error messages for JDBC or ODBC drivers. A terse error message is one without the SQLSTATE=*nnnn* prefix, where *nnnn* is the number of the error message.

By default, InfoMaker displays JDBC and ODBC error messages with the SQLSTATE prefix. To display error messages without the SQLSTATE prefix, set MsgTerse to 'Yes'.

Controls JDB JDBC
ODBC

Syntax MsgTerse='value'

Parameter	Description
<i>value</i>	<p>Specifies whether InfoMaker should display error messages without the SQLSTATE prefix. Values are:</p> <ul style="list-style-type: none"> • Yes Display error messages <i>without</i> the SQLSTATE prefix • No (Default) Display error messages <i>with</i> the SQLSTATE prefix

Default MsgTerse='No'

Usage You can set the MsgTerse parameter to 'Yes' to display shorter JDBC or ODBC error messages in InfoMaker. This might be useful if space on your screen is limited.

For example, suppose you are using the Data Pipeline in InfoMaker to pipe data to a **SQL** Anywhere ODBC database, and errors occur while you are executing the pipeline. If MsgTerse is set to 'No' (the default value), pipeline errors display in an Error dialog box *with* the SQLSTATE prefix (for example, SQLSTATE=23000).

If you specify MsgTerse='Yes' in the database profile of the **SQL** Anywhere destination database, the Data Pipeline displays terse ODBC error messages *without* the SQLSTATE prefix.

For instructions on using the Data Pipeline, see the InfoMaker *Users Guide*.

Examples To specify that InfoMaker should display terse error messages without the SQLSTATE prefix, select the Display Terse Error Messages check box on the System page in the Database Profile Setup dialog box.

NCharBind

Description Specifies whether InfoMaker binds input string parameters to the Char or NChar datatype.

Controls ADO.NET (Oracle.DataAccess.Client only)
O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12c
SNC SQL Native Client for Microsoft SQL Server

Syntax NCharBind=*value*

Parameter	Description
<i>value</i>	Specifies whether InfoMaker binds string input parameters to the Char or NChar datatype. Values are: <ul style="list-style-type: none">• 0 Binds string input parameters to the Char datatype.• 1 Binds string input parameters to the NChar datatype.

Default NCharBind=1 for SNC, NCharBind=0 for other interfaces

Usage **DisableBind must be set to 0**
For NcharBind to take effect, the DisableBind parameter must be set to 0. DisableBind=1 overrides the NcharBind setting.

Examples To specify that string arguments should be bound as the NChar datatype, select the NChar Bind box on the Transaction page in the Database Profile Setup dialog box.

See also BindSPInput
DisableBind
RPCRebind
UnicharBind

NCharLiteral

Description

Specifies whether the NChar literal replacement feature in the Oracle client is enabled. This feature replaces string literals on the client that are prefaced with the letter N with an internal format. The internal format is decoded to Unicode by the database server when the statement is executed.

When to specify NCharLiteral

You must specify the NCharLiteral parameter *before* connecting to the database.

Controls

O10 Oracle 10g

ORA Oracle 11g/12c

Syntax

NCharLiteral=*value*

Parameter	Description
<i>value</i>	<p>Specifies whether InfoMaker replaces string literals prefaced with the letter N in SQL command text. Values are:</p> <ul style="list-style-type: none"> • 'No' (Default) The text of any literal is encoded in the same character set as the rest of the statement on the client. • 'Yes' The text of any literal prefixed by N is replaced with an internal format.

Default

NCharLiteral='No'

Usage

The NCharLiteral database parameter requires Oracle 10.2 or higher on both the client and the database server.

By default, in a **SQL** statement, the text of any literal is encoded in the same character set as the rest of the statement. The character set on the client is determined by the client character set defined in NLS_LANG. When the statement is executed, the character set on the client is converted to the character set on the database server.

Data in string literals is lost in the conversion if the character set on the database server does not contain the characters used on the client. NChar string literals are most affected by this issue because they are designed to be independent of the character set on the database server.

To avoid this data loss, add the letter N before string literals that should be replaced with an internal format and set the NCharLiteral database parameter to 'Yes'. This setting causes the Oracle client to encode all literals prefixed with N in statements on the client with an internal format. The database server decodes the literals to Unicode when the statement is executed.

For example, when NCharLiteral is set to “Yes”, the string “some unicode data” in the following SQL statement is transferred from the client to the server with no data loss:

```
insert into table1 (id, ncharcol) values(1, N'some
unicode data')
```

Setting NCharLiteral to ‘Yes’ has no effect on DataWindow functions.

Examples

To specify that string literals prefixed with the letter N should be replaced with an internal format on the client, select the NChar Literal box on the General page in the Database Profile Setup dialog box.

NLS_Charset

Description

Specifies the client-side character set for the current environment handle.

Controls

O90 Oracle9i
 O10 Oracle 10g
 ORA Oracle 11g/12c

Syntax

NLS_Charset='value'

Parameter	Description
<i>value</i>	Specifies the client-side character set for the current environment handle. Values are: <ul style="list-style-type: none"> • Unicode (Default) The OCI client uses the UTF-16 character set. • Local The OCI client uses the current character set defined by the NLS_LANG parameter on the local computer. • ValidCharsetName Any valid character set name except AL16UTF16. To specify a value in the Database Profile Setup dialog box, type the name of the character set into the NLS Charset drop-down list box on the System page.

Default

NLS_Charset='Unicode'

Usage

When you specify a value for NLS_Charset, InfoMaker sets the OCI NLS_CHARACTERSET property in the current connection. NLS_CHARACTERSET is a database parameter that specifies the character set encoding used to store CHAR, VARCHAR2, LONG, and CLOB datatypes. The value you set corresponds to the character set defined by the NLS_LANG client-side parameter. NLS_LANG defines the character set encoding of text that OCI gets from or sends to applications in bind or define variables or as SQL statements to be executed by the server.

For the national character set NLS_NCHAR_CHARACTERSET, the Oracle database interfaces always use the UTF-16 character set. If you are using pooling with the ORA driver, and two Oracle connections are connected to the same Oracle server but use different character sets, the connections must reside in different connection or session pools. All pooling-related DBParm parameters must be set before the initial database connection.

For COM+ connections, you can use only Local.

Examples

To specify that the OCI client should use the current character set defined by the NLS_LANG parameter on the local computer to store string datatypes, select Local from the NLS Charset drop-down list on the System page in the Database Profile Setup dialog box.

See also

[StrByCharset](#)

NumbersInternal

Description

Specifies that numbers should be retrieved from the database using Oracle's internal 21-byte binary NUMBER datatype format instead of using OCI strings. The NumbersInternal parameter is relevant *only* when you are accessing an Oracle database configured with an EBCDIC character set or other non-ASCII character set.

When to specify NumbersInternal

You must specify the NumbersInternal parameter *before* connecting to the database.

Controls

O90 Oracle9*i*

O10 Oracle 10*g*

ORA Oracle 11*g*/12*c*

Syntax

NumbersInternal=*value*

Parameter	Description
<i>value</i>	Specifies that numbers should be retrieved using Oracle's internal 21-byte binary NUMBER datatype format. Values are: <ul style="list-style-type: none"> • 0 (Default) Do not retrieve numbers in internal format. • 1 Retrieve numbers in internal format.

Default

NumbersInternal=0

Usage In addition to specifying that numbers be retrieved from the database using Oracle's internal 21-byte binary NUMBER datatype format, the NumbersInternal parameter also provides an internal algorithm for deciphering the result.

Examples To specify that you want numbers to be retrieved using Oracle's internal format, select the Retrieve Numbers in Internal Format check box on the Syntax page in the Database Profile Setup dialog box.

NumericFormat

Description If supported by the DBMS or back-end database, setting NumericFormat tells the driver to do special formatting of numeric strings in SQL syntax. This formatting affects how InfoMaker generates numeric values in the SQL syntax it internally builds in reports and sends to your database.

Controls JDB JDBC
ODBC

Syntax The syntax you use depends on the back-end DBMS you are accessing and how you want to format the numeric string.

The following are typical syntax examples for Oracle databases that format a numeric string with a comma as the decimal separator. (See the Examples section for information about how InfoMaker generates numeric values in the SQL syntax it builds and sends to the database.)

In the InfoMaker development environment, the Database Profile Setup dialog box inserts special characters (quotes) where needed, so you can specify just the NumericFormat value (%s in this example).

IBM DB2 syntax If you are accessing an IBM DB2 database through the ODBC interface, use the following syntax for NumericFormat. Note the use of *one single quote* at the beginning and end of the string:

```
NumericFormat=' %s, %s '
```

Oracle JDBC or ODBC syntax If you are accessing an Oracle database through the JDBC or ODBC interface, use the following syntax for NumericFormat. Note the use of *three single quotes* at the beginning and end of the string:

```
NumericFormat=''' %s, %s '''
```

Parameter	Description
'	IBM DB2 syntax Type a single open quote. InfoMaker returns no open quote in the SQL syntax it builds and sends to the database, as required by IBM DB2 databases.
'''	Oracle, JDBC, or ODBC syntax Type three single open quotes. InfoMaker parses the second and third quotes as one single open quote in the SQL syntax it builds and sends to the database.
%s	Represents one or more digits to the <i>left of the decimal</i> in the numeric string. InfoMaker substitutes this value with the digits to the left of the decimal when it builds the SQL syntax.
,	Represents the decimal separator character (in this case a comma).
%s	Represents one or more digits to the <i>right of the decimal</i> in the numeric string. InfoMaker substitutes this value with the digits to the right of the decimal when it builds the SQL syntax.
'	IBM DB2 syntax Type one single closed quote. InfoMaker returns no closed quote in the SQL syntax it builds and sends to the database, as required by IBM DB2 databases.
'''	Oracle, JDBC, or ODBC syntax Type three single closed quotes. InfoMaker parses the first and second quotes as one single closed quote in the SQL syntax it builds and sends to the database.

Default

None

Usage

When to set NumericFormat In general, you should *not* need to set the NumericFormat parameter. Most back-end DBMSs do not require that the driver do special formatting of numeric strings in **SQL** syntax. However, some databases might require special formatting, such as an IBM DB2/MVS database server configured to use a comma as the decimal separator.

In these cases, setting NumericFormat allows you to generate numeric values with special formatting in the **SQL** syntax that InfoMaker builds in reports and sends to your database. For example, if the decimal separator for your DBMS is a comma, you might want to set NumericFormat as shown in the Examples section below to use a comma as the decimal delimiter in the **SQL** syntax sent to your database.

Examples

Example 1 (IBM DB2 syntax) This example shows how to specify that you want InfoMaker to generate two numeric values in the format *125,50* and *4,0*. InfoMaker uses the comma as a decimal separator in the **SQL** syntax it builds in reports and sends to an IBM DB2 database. Type the following in the Numeric Format box on the Syntax page in the Database Profile Setup dialog box:

```
%s,%s
```

What happens InfoMaker internally builds the following **SQL INSERT** statement in the report and sends the syntax to your database. InfoMaker returns no quotes in the **SQL** syntax.

```
INSERT INTO MYTABLE (a, b)
VALUES (125,50, 4,0)
```

Example 2 (Oracle, JDBC, or ODBC syntax) This example shows how to specify that you want InfoMaker to generate two numeric values in the format '125,50' and '4,0'. InfoMaker uses the comma as a decimal separator in the **SQL** syntax it builds in reports and sends to an Oracle database. Type the following in the Numeric Format box on the Syntax page in the Database Profile Setup dialog box:

```
%s,%s
```

What happens InfoMaker internally builds the following **SQL INSERT** statement in the report and sends the syntax to your database. InfoMaker returns single quotes in the **SQL** syntax.

```
INSERT INTO MYTABLE (a, b)
VALUES ('125,50', '4,0')
```

See also

DecimalSeparator

ObjectMode

Description

Allows InfoMaker to turn off the Oracle Call Interface (OCI) object mode. By default, InfoMaker sets the mode parameter of OCIInitialize(), the first OCI call in any OCI application, to OCI_OBJECT. When object mode is on, your application can define and use new database object types.

Controls

O90 Oracle9i
 O10 Oracle 10g
 ORA Oracle 11g/12c

Syntax

ObjectMode=*value*

Parameter	Description
<i>value</i>	Specifies whether object mode is enabled or not. Values are: <ul style="list-style-type: none"> Yes (Default) Use object mode. No Do not use object mode.

Default

Yes

Usage To turn ObjectMode off, clear the ObjectMode check box on the Connection page of the Database Profile Setup dialog box, or set ObjectMode to “No” in a script.

Examples To specify that you want ObjectMode disabled, clear the ObjectMode check box on the Connection page of the Database Profile Setup dialog box.

OJSyntax

Description Specifies how InfoMaker formats the **SQL** syntax for outer joins for the database back end you are accessing.

Controls ASE, SYC SAP Adaptive Server Enterprise
 I10 Informix
 IN9 Informix
 JDB JDBC
 ODBC
 OLE DB
 O90 Oracle9*i*
 O10 Oracle 10*g*
SNC SQL Native Client for Microsoft **SQL** Server

Syntax OJSyntax=*value*

Parameter	Description
<i>value</i>	Specifies how you want SQL syntax to be formatted. Values are: <ul style="list-style-type: none"> • ANSI_Escape Apply ANSI standards and enclose the outer joins in escape notation { oj ... } that is parsed by the driver and replaced with DBMS-specific grammar. • ANSI Apply ANSI standards. • PB Maintain rules that applied to InfoMaker 7.

Default OJSyntax=ANSI for IN9, **SNC**, ASE, and SYC, OJSyntax=ANSI_ESCAPE for JDBC, ODBC, and OLE DB, OJSyntax=PB for O90 and O10.

Usage All InfoMaker database interfaces provide support for ANSI **SQL**-92 outer join **SQL** syntax generation. InfoMaker supports both left and right outer joins in graphics mode and full outer and inner joins in syntax mode.

You must set the OJSyntax parameter to indicate the version of outer join **SQL** syntax you want InfoMaker to generate. For JDBC, ODBC, and OLE DB, the default is ANSI_Escape and can be reset to ANSI or PB (native). For IN9, **SNC**, ASE, and SYC, the default is ANSI and can be reset to PB. For O90 and O10, the default is PB, which means use Oracle native outer join syntax, and can be reset to ANSI.

Define outer joins in the SQL Select painter for portability When you define an outer join **SELECT** statement graphically in the **SQL** Select painter, the report stores the **SQL** in pseudocode. At runtime, the outer join syntax is generated based on the current OJSyntax parameter setting. This provides some degree of portability for reports among multiple DBMSs.

When you define an outer join **SELECT** statement in syntax mode, the report stores the **SQL** as syntax. This syntax is used without modification at runtime. The OJSyntax parameter setting does *not* affect the **SQL**.

Using native outer join syntax The option PB generates native outer join syntax. It is available for ODBC and OLE DB only if PBOuterJoin and PBOuterJoinOperator syntax entries are set in the appropriate SYNTAX section for your DBMS in the *Appeon\Shared\PowerBuilder\PBODB170.ini* file.

The PB option is available for JDBC only if PBOuterJoin and PBOuterJoinOperator syntax entries are set in the Windows registry in the appropriate key for your DBMS in the *HKEY_CURRENT_USER\Software\Sybase\PowerBuilder\17.0\pbjdbc* key. This key is not installed by default. See the *egreg.txt* file in *Appeon\Shared\PowerBuilder* for an example of a registry file you could execute to add or change InfoMaker JDBC settings for your DBMS.

When you migrate applications from InfoMaker 7 and earlier versions of InfoMaker, using ANSI outer join syntax might produce errors, depending on how the joins were defined in the painter. If a table is joined to multiple other tables with right outer joins, a valid ANSI outer join statement cannot be generated.

For more information about outer joins, see the section on using ANSI outer joins in the InfoMaker *Users Guide*.

OJSyntax does not apply to DIR For one database interface, DIR, the database connection always uses ANSI outer join **SQL** syntax.

Examples

To set the value of OJSyntax, select the appropriate value from the Outer Join Syntax drop-down list on the Syntax page in the Database Profile Setup dialog box.

OnlineIndex

Description

Specifies that the Database painter should use the **ONLINE** keyword when you create or drop an index on a table so that the index can be created or dropped without locking the table.

When to specify OnlineIndex

You must specify the OnlineIndex parameter *before* connecting to the database.

Controls

I10 Informix

Syntax

OnlineIndex='value'

Parameter	Description
<i>value</i>	Specifies that the Database painter should use the ONLINE keyword when you create or drop an index. Values are: <ul style="list-style-type: none"> • 0 The Database painter does not use the ONLINE keyword. • 1 The Database painter uses the ONLINE keyword.

Default

OnlineIndex=0

Usage

In IDS 10.0 and later, the **SQL** syntax of **CREATE INDEX** and **DROP INDEX** supports the **ONLINE** keyword to create or drop an index in an online environment where the database and its tables are continuously available. When you use the **ONLINE** keyword to create or drop an index, data definition language (DDL) operations execute without applying an exclusive lock on the table on which the specified index is defined.

If you use **CREATE INDEX ONLINE** to create an index on a table that other users are accessing, the index is not available until no users are updating the table. If you issue **DROP INDEX ONLINE** to drop an index, no users can reference the index, but concurrent data manipulation language (DML) operations can use the index until the operations terminate. Dropping the index is deferred until no users are using the index.

You can set the OnlineIndex static DbParameter on the System tab page in the Database Profile Setup dialog box for I10 connections to specify that the Database painter should use the **ONLINE** keyword when you create or drop an index.

Clustered index not supported

You cannot create a clustered index using online mode because it is not supported by IDS.

Examples To specify that the Database painter use the **ONLINE** keyword when you create or drop an index in the Database painter, select the Create or Drop Non-Clustered Indexes Without Dropping Tables check box on the System page in the Database Profile Setup dialog box.

PackageProcs

Description Specifies that the stored procedures and functions encapsulated in an Oracle database package should be appended to the lists of Oracle standalone stored procedures and functions displayed in the report and Database painters.

When to specify PackageProcs

You must specify the PackageProcs parameter *before* connecting to the database.

Controls O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12c

Syntax PackageProcs=*value*

Parameter	Description
<i>value</i>	Specifies that package-stored procedures and functions should be appended to the lists of stored procedures and functions. Values are: <ul style="list-style-type: none">• 0 (Default) Do not append package-stored procedures and functions.• 1 Append package-stored procedures and functions.

Default PackageProcs=0

Usage A package is an encapsulated collection of related program objects (such as procedures, functions, variables, and cursors) stored together in an Oracle database. Listing the objects contained in a package might impose a performance penalty on your Oracle database connection. When displayed in the Report painter, only those objects that contain a **REF CURSOR** or **SELECT** statement parameter are listed. When displayed in the Database painter, all objects are listed. The text source displayed is that of the entire package.

Examples To specify that you want Oracle package objects appended to the lists of stored procedures and functions, select the List Package Subprograms check box on the System page in the Database Profile Setup dialog box.

PacketSize (ODBC)

Description Specifies the network packet size in bytes when you access an ODBC data source in InfoMaker.

Many back-end DBMSs either do not support the PacketSize option or can return only the current network packet size. For information about whether the DBMS you are accessing supports PacketSize, see your DBMS documentation.

When to specify PacketSize

If your back-end DBMS supports it, you must specify the PacketSize parameter *before* connecting to the database.

Controls ODBC (if ODBC 2.0 or higher driver and back-end DBMS support this feature)

Syntax PacketSize=*value*

Parameter	Description
<i>value</i>	A 32-bit integer value that specifies the network packet size in bytes

Default The default value for PacketSize is the default for your back-end DBMS.

Usage If the PacketSize value you specify is larger than the maximum network packet size or smaller than the minimum network packet size, your ODBC driver substitutes the maximum or minimum value for the value you specified.

Examples To set the network packet size for an ODBC data source to 2048 bytes, type [2048](#) the Packet Size box on the Network page in the Database Profile Setup dialog box.

PacketSize (ASE, DIR, SNC, SYC)

Description When connecting to a database, specifies the packet size in bytes that you want the server to set for transferring data to and from your InfoMaker application. A **packet** is a fixed-size chunk of data for sending information over a network.

If the server has space limitations, it sets the packet size to less than the specified PacketSize value. Otherwise, it sets the size equal to the PacketSize value. The default value is 512 bytes.

When to specify PacketSize

You must specify the PacketSize parameter *before* connecting to the database.

Controls

ASE, SYC SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect
SNC SQL Native Client for Microsoft SQL Server

Syntax

PacketSize=*value*

Parameter	Description
<i>value</i>	A value specifying the packet size in bytes that a database server sets for transferring data to and from your application. The value must be a multiple of 512 bytes (default=512 bytes).

Default

PacketSize=512 (4096 for SNC)

Usage

When to set If your InfoMaker application sends or receives large amounts of text or image data from the server, setting the PacketSize value larger than the default 512 bytes might speed performance by causing fewer network read and write operations.

Adaptive Server Enterprise and DirectConnect Before setting PacketSize for use with an Adaptive Server Enterprise or DirectConnect database, you or your system administrator must set the following configuration variables on the server for PacketSize to take effect:

- **Additional netmem** Sets the maximum size of additional memory that can be used for network packets larger than the default size.
- **Maximum network packet size** Sets the maximum network packet size for all database users.

For instructions on setting these configuration variables, see your database documentation.

Examples

To specify that the database server should set the packet size equal to or less than 2048 bytes, type 2048 in the Packet Size box on the Network page in the Database Profile Setup dialog box.

PBCatalogOwner

Description

Specifies a nondefault owner for the extended attribute system tables. These five tables contain default extended attribute information for your database.

When you specify a PBCatalogOwner name that is different from the default owner for your DBMS, InfoMaker creates a new set of tables with the owner name you specify.

When to specify PBCatalogOwner

You must specify the PBCatalogOwner parameter *before* connecting to the database.

Controls

ASE, SYC SAP Adaptive Server Enterprise
 DIR SAP Sybase DirectConnect
 I10 Informix
 JDB JDBC
 ODBC
 OLE DB
 O90 Oracle9i
 O10 Oracle 10g
 ORA Oracle 11g/12c
 SNC SQL Native Client for Microsoft SQL Server

Syntax

PBCatalogOwner='owner_name'

Parameter	Description
<i>owner_name</i>	Specifies the owner of the extended attribute system tables. For DB2 databases If you use the DB2SYSPB.SQL script to create the extended attribute system tables in a DB2 database and replace all instances of PBOwner in the script with the name of a nondefault table owner, <i>owner_name</i> must be the same as the owner specified in the DB2SYSPB.SQL script.

Default

The default value for PBCatalogOwner depends on the DBMS you are accessing, as follows:

DBMS	PBCatalogOwner default value
Informix	PBCatalogOwner='informix'
JDBC	If a value for PBCatalogOwner is not specified in the database profile or in the registry, the default value is the user ID specified in the database profile.
ODBC	If a value for PBCatalogOwner is not specified in the database profile or in the PBODBN0 initialization file, the default value is the user ID specified in the database profile.
OLE DB	If a value for PBCatalogOwner is not specified in the database profile or in the registry, the default value is the user ID specified in the database profile.

DBMS	PBCatalogOwner default value
Oracle	PBCatalogOwner='SYSTEM'
SNC	PBCatalogOwner='dbo'
SAP Adaptive Server Enterprise	PBCatalogOwner='dbo'
SAP Sybase DirectConnect	PBCatalogOwner='sqlca.logid'

Usage

When to set When you specify a nondefault owner for the extended attribute system tables, you are in effect creating alternative tables. This is useful if you want to test new validation rules or display formats without overwriting the extended attributes currently in the default tables.

Informix databases For ANSI-compliant databases, the owner name that you specify must be unique but the table name does not have to be unique. You can create multiple sets of catalog tables prefaced with different user names. However, if the database is not ANSI-compliant, the table name must be unique, so that only one set of catalog tables can be created with an assigned owner name.

JDBC databases When you connect to a JDBC database and a value for PBCatalogOwner is set in both the database profile and the registry, the setting in the profile overrides the setting in the registry.

ODBC data sources When you connect to an ODBC data source and a value for PBCatalogOwner is set in both the database profile and the PBODB170 initialization file, the setting in the profile overrides the setting in the PBODB170 initialization file.

DB2 databases When you connect to a DB2 database, you can use the DB2SYSPB.SQL script to create the extended attribute system tables. If you use the DB2SYSPB.SQL script, keep the following in mind:

- You can edit the script to change all instances of PBOwner to another name, or leave the table owner as PBOwner in the script (the default).

Specifying SYSIBM is prohibited

DB2 prohibits you from specifying SYSIBM as the table owner.

- You can set the PBCatalogOwner parameter to the owner you specified in this script or to PBOwner if you did not edit the script.

This parameter cannot be set dynamically. The value set when the connection is made remains in effect until it is disconnected.

Examples This example shows how to create a new set of extended attribute system tables with the owner TEST. The names of the new tables have the prefix TEST, such as TEST.pbcatcol, TEST.pbcatedt, and so on. Type **TEST** in the PB Catalog Table Owner box on the System page in the Database Profile Setup dialog box.

PBMaxBlobSize

Description Specifies the maximum blob size that InfoMaker can read into memory.

When to specify PBMaxBlobSize

You must specify a value for the PBMaxBlobSize parameter *before* connecting to the database.

Controls

OLE DB
SNC SQL Native Client for Microsoft SQL Server

Syntax

PBMaxBlobSize=*value*

Default

PBMaxBlobSize=1024000

Usage

InfoMaker does not restrict the maximum blob size. Instead, the maximum blob size is determined by the machine on which the application is running. If the blob size exceeds the available memory on the machine on which the application is running, InfoMaker reads the blob in chunks if the data provider supports the ISequentialStream interface. If the blob size exceeds the default value and the data provider does not support the ISequentialStream interface, InfoMaker truncates it and reports an out-of-memory error. Use the PBMaxBlobSize parameter to specify larger maximum blob sizes.

Examples

To set the PBMaxBlobSize value to 200000, type **200000** in the Maximum In-Memory Blob Size box on the Transaction page in the Database Profile Setup dialog box.

PBMaxTextSize

Description

Specifies the maximum length of text or large value datatypes returned when you include them in a report.

You can set the PBMaxTextSize parameter if you want to include a long text string in a report without treating the text as a binary large object (blob) datatype.

Controls

SNC SQL Native Client for Microsoft SQL Server

Syntax

PBMaxTextSize='value'

Parameter	Description
value	The maximum length in bytes of text or large value datatypes returned when you include them in a report. The report can be created by a procedure. The range of valid values is from 10,000 bytes to 1,000,000 bytes.

Default

PBMaxTextSize=32767

Usage

The **SQL** Native Client OLE DB Provider for **SQL** Server automatically sets the **SQL** Server TEXTSIZE property to 2147483647 when connecting. The size of data fetched depends on the PBMaxTextSize parameter. If a result set includes a large value type, the blocking factor will be set to 1.

Examples

To have SQL Server return text or large datatypes that are up to 48,000 bytes long when you include them in a report, type 48000 in the Maximum String Size in DataWindow® box on the Transaction page in the Database Profile Setup dialog box.

PBNewSPInvocation

Description

Uses an alternative method to invoke a stored procedure.

Controls

ODBC

Syntax

PBNewSPInvocation='value'

Parameter	Description
value	Specifies whether the standard method or an alternative method is used to invoke a stored procedure. Values are: <ul style="list-style-type: none"> • No (Default) Use the standard method to invoke a stored procedure. • Yes Use the alternative method to invoke a stored procedure.

Default

PBNewSPInvocation='No'

Usage

Output parameters might not be returned when you use an embedded **SQL** command to call a stored procedure. You can set PBNewSPInvocation to 'Yes' to use an alternative method to invoke a stored procedure. The behavior of the InfoMaker ODBC driver when this parameter is set is consistent with the default behavior of the OLE DB and JDBC drivers.

If `PBNewSPInvocation` is set to 'Yes', the alternative method is used when you retrieve data into a report that uses a stored procedure. This parameter has no effect when you use RPC to invoke a stored procedure.

When `PBNewSPInvocation` is set to 'Yes', the values of the `PBUseProcOwner` and `CallEscape` parameters are ignored.

Examples

To set the parameter for all connections, add the following line to every relevant section (such as `;IBM DB2/NT 2.1 DB2CLI` for a DB2 connection on Windows) in your `PBODBI70.ini` file:

```
PBNewSPInvocation='Yes'
```

For more information about editing `PBODBI70.ini`, see the Appendix in *Connecting to Your Database*.

To obtain the value of the stored procedure's output parameter, use the `OUTPUT` or `OUT` keyword. For example:

```
DECLARE sp_test PROCEDURE FOR SP1 VAR0=:ARGIN,
VAR1=:ARGOUT OUTPUT USING SQLCA;
```

If the stored procedure contains result sets, you must fetch the result sets first. If the stored procedure has a return value and you want to obtain it, use the format `RC=SP1`:

```
DECLARE sp_test PROCEDURE FOR RC=SP1 VAR0=:ARGIN,
VAR1=:ARGOUT OUTPUT USING SQLCA;
```

See also

[DefaultProcOwner](#)

PBTrimCharColumns

Description

Specifies whether InfoMaker should trim trailing spaces from data values retrieved from the following datatypes: Char, Char for Bit Data, VarChar, and VarChar for Bit Data.

Controls

ODBC
OLE DB

Syntax

`PBTrimCharColumns=value`

Parameter	Description
<i>value</i>	Specifies whether InfoMaker should trim trailing spaces from data of type Char, Char for Bit Data, and VarChar for Bit Data. Values are: <ul style="list-style-type: none"> NO (Default) Do not trim trailing spaces. YES Trim trailing spaces.

Default	'NO'
Usage	<p>This parameter can only be set in the <i>PBODBI70.ini</i> file. For ODBC, you can set the TrimSpaces parameter in the Database Profile Setup dialog box to perform the same function.</p> <p>By default, InfoMaker trims spaces from the following datatypes: Char, Char for Bit Data, VarChar, and VarChar for Bit Data.</p> <p>If your DBMS makes a distinction between Char data with trailing spaces and Char data without trailing spaces when evaluating a WHERE clause expression, you might receive the message <i>Row changed between retrieve and update</i> when your form's update properties are set to "Key and updateable columns." To prevent this, change your form's update properties. In embedded SQL, you can check <code>Sqlca.Sqlnrows</code> after each update to determine if the update took place. Avoid using Char data columns in the WHERE clause of an UPDATE or DELETE statement when <code>PBTrimCharColumns='YES'</code>.</p>
Examples	<p>To specify that InfoMaker should trim trailing spaces, add the following line to the section for the database you are accessing:</p> <pre>PBTrimCharColumns='YES'</pre>
See also	TrimSpaces

PBUseProcOwner

Description	<p>When you access a database through the ODBC interface and define a report that uses a stored procedure as its data source, <code>PBUseProcOwner</code> specifies whether InfoMaker should qualify the stored procedure with the owner name in the SQL EXECUTE statement passed to the driver.</p> <p>InfoMaker qualifies the stored procedure with an owner only if the owner associated with the stored procedure is different from the ID of the current user (the developer building the report or the user running the application containing the report).</p>
Controls	ODBC
Syntax	<code>PBUseProcOwner='value'</code>

Parameter	Description
<i>value</i>	<p>Specifies whether InfoMaker should qualify the stored procedure with its owner name in the SQL EXECUTE statement built by the report and passed to the driver. Values are:</p> <ul style="list-style-type: none"> • Yes If the owner associated with the stored procedure is different from the current user ID, InfoMaker qualifies the stored procedure with its owner name in the SQL EXECUTE statement and passes this information to the driver. This allows users to execute stored procedures they do not own. For example: <pre>EXECUTE FRAN.MYPROCEDURE</pre> • No (Default) InfoMaker does not qualify the stored procedure with its owner name in the SQL EXECUTE statement passed to the driver. For example: <pre>EXECUTE MYPROCEDURE</pre>

Default

PBUseProcOwner='No'

Usage

Determining the PBUseProcOwner value InfoMaker searches the following in this order to determine the PBUseProcOwner value:

- 1 The section for your database profile in the InfoMaker initialization file.
- 2 The section for your ODBC driver in the *PBODBI70* initialization file.

If InfoMaker does not find a PBUseProcOwner value in these locations, it defaults to a value of “No”.

If DBA owns the SQL Anywhere® stored procedure DBA (database administrator) is a reserved word in SQL Anywhere syntax.

If you define a report with a SQL Anywhere stored procedure as its data source and DBA owns the stored procedure, the painter passes the following SQL EXECUTE statement to the ODBC driver if PBUseProcOwner is set to “Yes”:

```
EXECUTE DBA.MYPROCEDURE
```

This statement generates a syntax error because it includes the DBA reserved word.

If DBA owns the SQL Anywhere stored procedure you are using, you can avoid this syntax error by setting PBUseProcOwner to No so that InfoMaker does not qualify the stored procedure with DBA.

In some situations, however, you *must* qualify the stored procedure with the DBA owner. For example, the DBA might want to grant execute permission to another user ID. In this case, you can avoid errors by editing the **SQL EXECUTE** syntax to enclose DBA in quotes, like this:

```
EXECUTE "DBA".MYPROCEDURE
```

Examples

To specify that InfoMaker should qualify the stored procedure with its owner name in the **SQL EXECUTE** statement, select the Qualify Stored Procedures With Owner Name check box on the Transaction page in the Database Profile Setup dialog box.

PersistEncrypted

Description

Specifies whether the data source you are accessing through the OLE DB interface is allowed to save your encrypted password.

When to specify PersistEncrypted

You must specify the PersistEncrypted parameter *before* connecting to the database.

Controls

OLE DB

Syntax

PersistEncrypted=*value*

Parameter	Description
<i>value</i>	Specifies whether the data source can save your encrypted password. Values are: <ul style="list-style-type: none">• True Tells the data source it can save your password.• False (Default) Tells the data source it cannot save your password.

Default

PersistEncrypted='False'

Examples

To tell the data source you are accessing through OLE DB that it can save your password, select the Persist Encrypted check box on the Security page in the Database Profile Setup dialog box.

See also

DataLink
MaskPassword
EncryptPassword
PersistSensitive

PersistSensitive

Description Specifies whether the data source you are accessing through the OLE DB interface is allowed to save sensitive authentication information, such as a password, along with other authentication information.

When to specify PersistSensitive

You must specify the PersistSensitive parameter *before* connecting to the database.

Controls

OLE DB

Syntax

PersistSensitive='value'

Parameter	Description
<i>value</i>	Specifies whether the data source can save your authentication information. Values are: <ul style="list-style-type: none"> • True Tells the data source it can save your authentication information. • False (Default) Tells the data source it cannot save your authentication information.

Default

PersistSensitive='False'

Examples

To tell the data source you are accessing through OLE DB that it can save your authentication information, select the Persist Security Info check box on the Security page in the Database Profile Setup dialog box.

See also

MaskPassword
 EncryptPassword
 PersistEncrypted

PoolCreator

Description Specifies the user name used to create a connection or session pool.

Controls

ORA Oracle 11g/12c

Syntax

PoolCreator =*value*

Parameter	Description
<i>value</i>	String to specify the name of the connection or session pooling creator.

Default

None.

Usage Use in conjunction with the PoolPWD DBParm. If no value is provided for this DBParm, the LogID value is used to create the connection or session pool.

Examples To set the name of a connection pool creator to Scott, select Connection Pooling from the Pooling Type drop-down list on the Pooling page in the Database Profile Setup dialog box, type **Scott** in the Pool Creator text box, and Scott's password in the Password text box on the same page.

See also [Pooling](#)
[PoolPwd](#)

Pooling

Description Specifies the type of pooling to use with an Oracle database.

Controls ORA Oracle 11g/12c

Syntax Pooling =*value*

Parameter	Description
<i>value</i>	Specifies whether to use pooling, and if so, what kind of pooling. Values are: <ul style="list-style-type: none">• Session For session pooling.• Connection For connection pooling.• None (Default) For no pooling.

Default No pooling.

Usage With the ORA driver you specify the type of pooling you want with the Pooling DBParm. The O90 and O10 database drivers that you can use in InfoMaker to connect to the 9.x and 10.x versions of the Oracle DBMS support connection pooling with the DBParm parameter CNNPool. For backward compatibility purposes, the CNNPool parameter is also supported by the ORA driver. However, if you set the Pooling parameter with the ORA driver, the CNNPool parameter is ignored.

If you are using connection or session pooling, the proxy user name is the connection or session pooling creator (which you can provide in the PoolCreator and PoolPwd DBParm parameters), and the Transaction object's LogID is ignored.

If you select session pooling, you can enter a value for the SessionHomogeneous DBParm to authenticate all sessions in the pool with the user name and password in effect when the session pool was created. No proxy session can be created if pooling is set to homogeneous session mode.

Examples To set up a connection pool, select Connection Pooling from the Pooling Type drop-down list on the Pooling page in the Database Profile Setup dialog box, type [Scott](#) in the Pool Creator text box, and Scott's password in the Password text box on the same page.

See also [PoolCreator](#)
[PoolPwd](#)

PoolPwd

Description Specifies the password used to create a connection or session pool.

Controls ORA Oracle 11g/12c

Syntax `PoolCreator =value`

Parameter	Description
<i>value</i>	String to specify the name of the connection or session pooling creator.

Default None

Usage Use in conjunction with the PoolCreator DBParm. If no value is provided for this DBParm, the LogPass value of the Transaction object is used to create the connection or session pool.

Examples To set the password of a connection pool creator named Scott, select Session Pooling from the Pooling Type drop-down list on the Pooling page in the Database Profile Setup dialog box, type [Scott](#) in the Pool Creator text box, and type [mypass](#) in the Password text box on the same page.

See also [PoolCreator](#)
[Pooling](#)

Properties

Description Sets properties specific to the particular JDBC driver you are using to connect to the database.

Controls JDB JDBC

Syntax `Properties='property_value'`

Default None

Usage The Driver-Specific Properties box allows you to set properties specific to a particular driver.

For information about the properties supported by your JDBC driver, see the vendor's documentation.

Define User ID and Password

If properties are defined, you *must* also define the user ID and password in the properties box.

Examples To set a property for the SAP Sybase jConnect driver, type the following in the Driver-Specific Properties box on the Connection page in the Database Profile Setup dialog box:

```
SQLINITSTRING=set TextSize 32000;  
user=sa;password=manager
```

See also Driver
URL

ProtectionLevel

Description Specifies the level of protection applied to data sent between InfoMaker and the data server through the OLE DB data provider. This parameter applies only to network connections other than Remote Procedure Call (RPC) connections. Similar levels of protection can be specified for authenticated RPC connections.

When to specify ProtectionLevel

You must specify the ProtectionLevel parameter *before* connecting to the database.

Controls OLE DB

Syntax ProtectionLevel='value'

Parameter	Description
<i>value</i>	<p>Specifies the level of protection applied to data sent between InfoMaker and the data server. Values are:</p> <ul style="list-style-type: none"> • Not set No level of protection is selected. • Call Authenticates the source of the data at the beginning of each request from the client to the server. • Connect Authenticates only when the client establishes the connection with the server. • None Performs no authentication of data. • Packet Authenticates that all data received is from the client. • Packet (Integrity) Authenticates that all data received is from the client and that it has not been changed in transit. • Packet (Privacy) Authenticates that all data received is from the client, that it has not been changed in transit, and that it protects the privacy of the data by encrypting it.

Default Not set

Examples To set a level of protection for the data sent between InfoMaker and the data server, on the Security page in the Database Profile Setup dialog box, select Connect from the Protection Level drop-down list.

See also [DataLink](#)

Provider

Description Identifies the data provider you want to use to connect to your data source.

When to specify Provider

You must specify the Provider parameter *before* connecting to the database.

Controls OLE DB
SNC SQL Native Client for Microsoft SQL Server

Syntax `Provider='provider_name'`

Default None

Usage Select a data provider from the list of installed data providers displayed in the Provider drop-down list. For example, if you are using Microsoft's OLE DB Provider for ODBC, select MSDASQL as the Provider value. If you are using Microsoft's OLE DB Provider for SQL Server, select SQLOLEDB as the Provider value.

For more information, see the documentation provided by your OLE DB data provider.

SNC SQL Native Client driver

The Provider DBParm parameter for the Microsoft SQL Native Client (SNC) interface allows you to select the SNC version that you want to use for a database connection. You can set this parameter in script to `SQLNCLI` (for the SNC 9.0 driver that connect to SQL Server 2005) or to `SQLNCLI10` (for the SNC 10.0 driver that connects to SQL Server 2008). Otherwise, you can select one of these providers on the Connection tab of the Database Profile Setup dialog box for the SNC interface.

If you do not set or select a provider, the default selection is `SQLNCLI` (SNC 9.0 for SQL Server 2005). This allows existing SNC interface users to be able to migrate to the current release of InfoMaker without any modifications. If InfoMaker fails to connect with the `SQLNCLI` provider, it will attempt to connect to `SQLNCLI10` provider. However, if you explicitly set the provider and the connection fails, InfoMaker displays an error message.

Examples

Example 1 To use the Microsoft OLE DB Provider for ODBC to connect to the PB Demo DB, select MSDASQL from the Provider drop-down list on the Connection page in the Database Profile Setup dialog box for OLE DB.

Example 2 To use the Microsoft OLE DB Provider for Oracle to connect to an Oracle 12c database, select MSDAORA from the Provider drop-down list on the Connection page in the Database Profile Setup dialog box.

Example 3 To use the Microsoft SNC Provider to connect to a Microsoft SQL Server 2008 database, select `SQLNCLI10` from the Provider drop-down list on the Connection page in the Database Profile Setup dialog box.

See also

[DataLink](#)
[DataSource](#)

ProviderString

Description

A string containing provider-specific extended connection information. Use of this database parameter requires that you know how this string will be interpreted and used by the provider. You should use this parameter only for provider-specific connection information that cannot be explicitly described by other database parameters.

When to specify ProviderString

You must specify the ProviderString parameter *before* connecting to the database.

Controls

OLE DB
SNC SQL Native Client for Microsoft **SQL** Server

Syntax

ProviderString='value'

Default

None

Usage

OLE DB applications can initialize data source objects using two methods: **IDBInitialize::Initialize** and **IDataInitialize::GetDataSource**.

Using **IDBInitialize::Initialize**, a provider string can be used to initialize connection properties by setting the DBPROP_INIT_PROVIDERSTRING property in the DBPROPSET_DBINIT property set. An initialization string can also be passed to the **IDataInitialize::GetDataSource** method to initialize connection properties.

For the OLE DB interface, the ProviderString parameter is used with **IDBInitialize::Initialize**. For the **SNC** interface, the ProviderString parameter is used with **IDataInitialize::GetDataSource**.

Both methods initialize the same OLE DB connection properties, but they use different sets of keywords. For lists of keywords, see the information about OLE DB provider connection string keywords in the Microsoft documentation.

Examples

Since Microsoft **SQL** Server supports multiple instances of a database on a single server, you must identify the specific database to which you want to connect by entering the database name. For the **SNC** interface, you should set the Database parameter on the Connection page in the Database Profile Setup dialog box. For OLE DB, you can use the ProviderString parameter.

To identify a specific Microsoft **SQL** Server database named **ts3**, enter the following in the Extended Properties box on the Connection page in the Database Profile Setup dialog box for OLE DB:

```
Database=ts3
```

See also

URL

PWDIALOG

Description

Controls whether a Password Expired dialog box displays in an application at runtime if a user's password has expired. This parameter has no effect in InfoMaker.

When PWDIALOG is set to 1, the Password Expired dialog box prompts users to change their passwords if they attempt to log in to the database with an expired password. By default, PWDIALOG is set to 0 to specify that the Password Expired dialog box does not display in your application at runtime.

The setting of PWDIALOG affects applications only at runtime. It has no effect in the development environment because, regardless of the PWDIALOG setting, the Change Password dialog box displays in the development environment to prompt users to change an expired password.

When to specify PWDIALOG

You must specify a value for PWDIALOG *before* connecting to the database.

Controls

ASE, SYC SAP Adaptive Server Enterprise
 O90 Oracle9i
 O10 Oracle 10g
 ORA Oracle 11g/12c

Syntax

PWDIALOG=*value*

Parameter	Description
<i>value</i>	Specifies whether the Password Expired dialog box displays in an application at execution time to prompt the user to change an expired login password. Values are: <ul style="list-style-type: none"> • 0 (Default) Do not display the Password Expired dialog box at execution time. • 1 Display the Password Expired dialog box at execution time to prompt the user to change an expired password.

Default

PWDIALOG=0

Usage

When to use Setting PWDIALOG to 1 to display the Password Expired dialog box in your application provides a convenient way for you to notify your users that a password has expired and allow them to change it.

What happens When the Password Expired dialog box displays in your application at runtime, it notifies users that the password for their login ID has expired and prompts them to supply a new password. For example, for Adaptive Server Enterprise, the `sp_password` system stored procedure runs to set the new password. Once the password has been changed, the database connection succeeds.

If the user clicks Cancel to close the Password Expired dialog box without changing the password, the database connection fails and a message displays indicating that the password has expired.

PWEncrypt

Description

PWEncrypt specifies whether you want Open Client to automatically encrypt your password when connecting to an SAP Adaptive Server Enterprise database in InfoMaker.

When to specify PWEncrypt

You must specify the PWEncrypt parameter *before* connecting to the database.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

PWEncrypt='value'

Parameter	Description
<i>value</i>	Specifies whether you want the Open Client software to encrypt your password. Values are: <ul style="list-style-type: none"> • Yes (Default) Tells Open Client to encrypt the password by setting the <code>CS_SEC_ENCRYPTION</code> connection property to <code>CS_TRUE</code>. • No Tells Open Client not to encrypt the password by setting the <code>CS_SEC_ENCRYPTION</code> connection property to <code>CS_FALSE</code>.

Default

PWEncrypt='Yes'

Examples

To tell Open Client not to encrypt your password when connecting to an SAP Adaptive Server Enterprise database in InfoMaker, clear the Encrypt Password check box on the Network page in the Database Profile Setup dialog box.

QualifyPublic

Description

Specifies that the PUBLIC qualifier prepended to Oracle synonyms belonging to the public schema or user group is retained in the **SQL** Select table list.

When to specify QualifyPublic

You must specify the QualifyPublic parameter *before* connecting to the database.

Controls

O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12c

Syntax

QualifyPublic=*value*

Parameter	Description
<i>value</i>	Specifies that the PUBLIC qualifier should be retained in the SQL Select table list. Values are: <ul style="list-style-type: none">• 0 (Default) Do not retain the PUBLIC qualifier.• 1 Do retain the PUBLIC qualifier.

Default

QualifyPublic=0

Usage

InfoMaker's default behavior has been to discard the PUBLIC qualifier so that the object reference is generalized in the generated **SQL** statement, facilitating the deployment of an application from a development database instance to a production database. However, in certain reports, the absence of the PUBLIC qualifier breaks the association of the synonym with its extended attributes, preventing these attributes from being used. The QualifyPublic parameter allows you to specify whether the PUBLIC qualifier should be retained.

Examples

To specify that you want the PUBLIC qualifier to be retained in the **SQL** Select table list, select the Qualify Public Synonyms check box on the System page in the Database Profile Setup dialog box.

Release

Description

Specifies what version of SAP Sybase Open Client Client-Library (CT-Lib) software is in use on the client workstation.

When to specify Release

You must specify a value for Release *before* connecting to the database.

Controls ASE, SYC SAP Adaptive Server Enterprise

Syntax Release='value'

Parameter	Description
<i>value</i>	<p>Specifies the version of Open Client Client-Library your application uses.</p> <p>The value corresponds to the value of the CS_VERSION property that is used to allocate a context structure for the client. Multiple values that specify the same client context are provided for backwards compatibility. Release number 15 or higher for ASE. Release number 15 or lower for SYC:</p> <ul style="list-style-type: none"> • 10.x CS_VERSION_100 • 11 CS_VERSION_110 • 11.5 CS_VERSION_110 • 12 CS_VERSION_110 • 12.5 CS_VERSION_125 • 12.5.1 CS_VERSION_125 • 15 CS_VERSION_15 • 15.5 CS_VERSION_155

Default Release='15' for ASE, Release='11' for SYC

Usage The Release parameter must correspond to the version of Open Client software installed on the client workstation. For example, do not specify 12.5 or 12.5.1 if your Open Client version is 12.0, even if your Adaptive Server version is 12.5 or 12.5.1.

To use Adaptive Server 15, you must install Open Client version 15 on the client computer and set the Release parameter to 15 to establish an Open Client 15 client context.

The Open Client context is allocated by the first *PBSYC170.DLL* database connection. This context acts as the parent context for all subsequent *PBSYC170.DLL* connections. Therefore, you must specify the same value for the Release parameter for all your connections.

If the client library is not loaded, the version that corresponds to the value of Release is loaded. If there is no active client context, *PBSYC170.DLL* connects using the value of the Release parameter. If there is an active client context, the new connection fails if the value of the Release parameter is higher than the current client context. The new connection succeeds if the value of the Release parameter is the same as or lower than the current client context.

The SYC driver links to the appropriate version of the client libraries dynamically and the Open Client context is released when all connections are closed. If you open multiple connections, the first Open Client context established is used for all of them. If you need to establish a new Open Client context in the development environment, close all open connections and establish a new connection with the Release parameter set to the context you require.

During each database login, *PBSYC170.DLL* automatically determines the version of Adaptive Server being accessed. It customizes its behavior to optimize performance and features for the combination of the Adaptive Server version and the Open Client context specified in the Release parameter. Specifying a value for Release that does not correspond to the Open Client software on the client can cause unpredictable results.

The values 12.5 and 12.5.1 both open an Open Client 12.5 context. However, you should always specify Release='12.5' if you are using Open Client 12.5 and Release='12.5.1' if you are using Open Client 12.5.1. This ensures that *PBSYC170.DLL* correctly handles the following scenarios that require Open Client 12.5.1 and Adaptive Server 12.5.1:

- Use the **Date** and **Time** datatypes introduced in Adaptive Server 12.5.1 in RPC calls that explicitly call for these datatypes in the stored procedure argument list.
- Use the **Date** and **Time** datatypes in `Update where current of` and `Delete where current of` statements.

Retrieval, insert, update, and delete processing work correctly against **Date** and **Time** datatypes using any Open Client software and Adaptive Server 12.5.1 or later. In the Database painter, the **Date** and **Time** datatypes display in the list of metadata types when you are connected to an Adaptive Server 12.5.1 server in any Open Client context.

Certain other features are supported only when you access a specified version of a **SQL** Server 10/11 or Adaptive Server Enterprise database using its associated Open Client software. For example, you must:

- Set the Release parameter to 11 or higher and use Open Client 11.x or higher and Adaptive Server 11.x or higher to take advantage of network-based security and directory services in your application.
- Set Release to 12.5 and use Open Client 12.5 or higher and Adaptive Server 12.5 or higher to access **Char** and **VarChar** columns with more than 255 characters.

- Set Release to 15 and use Open Client 15 or higher and Adaptive Server 15 or higher to access the **UniText** and 64-bit integer (**BigInt**) SQL datatypes added in version 15 of Adaptive Server.
- Set Release to 15.5 and use Open Client 15.5 or higher to access the **BIGTIME** and **BIGDATETIME** SQL datatypes added in version 15.5 of Adaptive Server.

Examples

To specify that your InfoMaker application accesses an Adaptive Server Enterprise 15 database using an Open Client Client-Library 15 context, select 15 from the Release drop-down list on the Connection page in the Database Profile Setup dialog box.

Request**Description**

Specifies whether to allocate new transaction resources each time the client application sends a request and then release those resources after each request.

Controls

DIR SAP Sybase DirectConnect (applies only to Access Service for DB2/MVS)

Syntax

Request=*value*

Parameter	Description
<i>value</i>	Specifies whether you want to release transaction resources after each request. Values are: <ul style="list-style-type: none"> • 0 (Default) Maintain resources for the duration of the database connection. • 1 Release resources after each request.

Default

Request=0

Usage

Requirements for using the Request parameter Setting the Request parameter to 1 to release resources has an effect only when you do both of the following:

- Set the AutoCommit database preference to True to specify that InfoMaker should issue SQL statements outside the scope of a transaction. (See the description of **AutoCommit**.)
- Specify the value for Request *before* connecting to a database.

What happens When you set the Request parameter to 1, transaction resources are allocated for each request and released when the request finishes. This might slow the performance of your application, but it allows more simultaneous users of the system.

Examples To specify that you want to release resources after each request, select the Release Transaction Resources After Each Request check box on the Transaction page in the Database Profile Setup dialog box.

See also [AutoCommit](#)

RPCRebind

Description Specifies whether you want InfoMaker to rebind Remote Procedure Call (RPC) parameters.

When to specify RPCRebind

If your back-end DBMS supports it, you must specify the RPCRebind parameter *before* connecting to the database.

Controls ODBC

Syntax `RPCRebind=value`

Parameter	Description
<i>value</i>	Specifies whether you want InfoMaker to rebind RPC parameters. Values are: <ul style="list-style-type: none">• 0 (Default) Use the bound variable to determine all required binding information.• 1 Rebind the parameters and use the parameter information returned from the database to bind the parameter.

Default RPCRebind=0

Usage For those DBMSs that support RPC calls, InfoMaker binds the parameters for the call based on the size of the variables bound to the parameters.

Some drivers require rebinding of the parameters so the parameter size (as returned from the back-end database) is used instead of the variable size. Failure to do this might result in an error or truncation for string parameters. However, some drivers always expect the binding to reflect the variable size. The RPCRebind parameter allows you to specify whether you want to rebind the parameters when RPCs are executed.

Examples To specify that InfoMaker should rebind RPC parameters, select the RPC Rebind check box on the Transaction page in the Database Profile Setup dialog box.

RTConnBalancing

Description Supports the runtime connection load balancing feature.

Controls ORA Oracle 11g/12c

Syntax RTConnBalancing =*value*

Parameter	Description
<i>value</i>	Specifies whether the load balancing feature is used. Values are: <ul style="list-style-type: none"> • True (Default) The load balancing feature is used if session pooling is set. • False The load balancing feature is not used.

Default True, but this value is ignored if session pooling is not set.

Usage Runtime connection load balancing routes work requests to the sessions in a session pool that best serve the work. Runtime connection load balancing is enabled by default when an Oracle 11.1 or higher client is connected to a release 10.2 or higher Oracle server using OCI session pooling.

Examples To disable the load balancing feature for a session pool, select Session Pooling from the Pooling Type drop-down list on the Pooling page in the Database Profile Setup dialog box and clear the Runtime Connection Load Balancing for Session Pooling check box on the same page.

See also [Pooling](#)

Scroll

Description Specifies whether you want to use a scroll cursor when connecting to an Informix database in InfoMaker. When you fetch rows in an Informix table, using a scroll cursor enables you to fetch the next row, previous row, first row, or last row.

By default, InfoMaker does not use scroll cursors in an Informix database connection.

Controls I10 Informix
IN9 Informix

Syntax Scroll=*value*

Parameter	Description
<i>value</i>	<p>Specifies whether you want to use a scroll cursor when connecting to an Informix database in InfoMaker. Values are:</p> <ul style="list-style-type: none"> • 0 (Default) Do not use a scroll cursor. • 1 Use a scroll cursor.

Default

Scroll=0

Examples

To specify that you want to use a scroll cursor when connecting to an Informix database in InfoMaker, select the Use A Scroll Cursor check box on the Transaction page in the Database Profile Setup dialog box.

Sec_Channel_Bind

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, Sec_Channel_Bind is one of several parameters that support login authentication for network-based security services. (For other login authentication parameters, see the See Also section.)

Sec_Channel_Bind controls whether your connection's security mechanism performs channel binding. When Sec_Channel_Bind is set to 1, both SAP Sybase Open Client Client-Library (CT-Lib) and the server provide a network channel identifier to the security mechanism before connecting. The channel identifier contains the network addresses of the client and server.

When Sec_Channel_Bind is set to 0 (the default), no channel binding is performed.

You must specify a value for Sec_Channel_Bind *before* connecting to the database.

Using third-party security mechanisms

For information about the third-party security mechanisms and operating system platforms that Appeon has tested with Open Client security services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

Sec_Channel_Bind=*value*

Parameter	Description
<i>value</i>	<p>Specifies whether your connection's security mechanism performs channel binding. Values are:</p> <ul style="list-style-type: none"> • 0 (Default) Do <i>not</i> perform channel binding. You can also specify 'No' or 'False' to set this value. • 1 Perform channel binding. Both CT-Lib and the server provide a channel identifier to the connection's security mechanism. You can also specify 'Yes' or 'True' to set this value.

Default

Sec_Channel_Bind=0

Usage

Not supported with CyberSafe Kerberos Sec_Channel_Bind is *not supported* if your security mechanism is CyberSafe Kerberos.

Set Release parameter For this parameter to take effect, you *must* also set the *Release* parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the *Release* parameter for more information.

Requirements for use To use Sec_Channel_Bind or any other parameter supporting Open Client, you must meet certain requirements for using security services in your InfoMaker application. For details, see “Requirements for using Open Client security services” in *Connecting to Your Database*.

Corresponding CT-Lib connection property Specifying a value for Sec_Channel_Bind sets the corresponding SAP Sybase CT-Lib connection property named CS_SEC_CHANBIND.

Examples

To specify that your connection's security mechanism performs channel binding, select the Enable Channel Binding check box on the Security page in the Database Profile Setup dialog box.

See also

[Sec_Cred_Timeout](#)
[Sec_Delegation](#)
[Sec_Keytab_File](#)
[Sec_Mechanism](#)
[Sec_Mutual_Auth](#)
[Sec_Network_Auth](#)
[Sec_Server_Principal](#)
[Sec_Sess_Timeout](#)

Sec_Confidential

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, Sec_Confidential is one of several parameters that support per-packet security for network-based security services. (For other per-packet security parameters, see the See Also section.)

Sec_Confidential controls whether transmitted data is encrypted. When Sec_Confidential is set to 1, all requests sent to the server and all results returned by the server are encrypted.

When Sec_Confidential is set to 0 (the default), transmitted data is not encrypted.

You must specify a value for Sec_Confidential *before* connecting to the database in InfoMaker.

Using third-party security mechanisms

For information about the third-party security mechanisms and operating system platforms that Apeon has tested with Open Client security services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

Sec_Confidential=*value*

Parameter	Description
<i>value</i>	<p>Specifies whether transmitted data is encrypted. Values are:</p> <ul style="list-style-type: none"> 0 (Default) Do <i>not</i> encrypt transmitted data. You can also specify 'No' or 'False' to set this value. 1 Encrypt transmitted data. Requests sent to the server and results returned by the server are encrypted. You can also specify 'Yes' or 'True' to set this value.

Default

Sec_Confidential=0

Usage

When to use Encryption can protect your data if you are sending it over a public network to a nonsecure server. In a networked environment, you might want to set Sec_Confidential to 1 to ensure that all requests sent to the server and all results returned by the server are encrypted.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the [Release](#) parameter for more information.

Requirements for use To use Sec_Confidential or any other parameter supporting Open Client security services, you must meet certain requirements for using security services in your InfoMaker application. For details, see “Requirements for using Open Client security services” in *Connecting to Your Database*.

Corresponding CT-Lib connection property Specifying a value for Sec_Confidential sets the corresponding SAP Sybase CT-Lib connection property named CS_SEC_CONFIDENTIALITY.

Examples

To specify that transmitted data is encrypted, select the Encrypt All Results check box on the Security page in the Database Profile Setup dialog box.

See also

Release
Sec_Data_Integrity
Sec_Data_Origin
Sec_Replay_Detection
Sec_Seq_Detection

Sec_Cred_Timeout

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, Sec_Cred_Timeout is one of several parameters that support login authentication for network-based security services. (For other login authentication parameters, see the See Also section.)

Some security mechanisms allow applications to set credential timeout values for connections that use network-based login authentication.

Sec_Cred_Timeout specifies the number of seconds remaining before a user’s network credentials expire and become invalid. Users obtain network credentials when they log in to the network.

By default, Sec_Cred_Timeout specifies that there is no credential timeout limit—the credentials do not expire.

You must specify a value for Sec_Cred_Timeout *before* connecting to the database in InfoMaker.

Using third-party security mechanisms

For information about the third-party security mechanisms and operating system platforms that Apeon has tested with Open Client security services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

Sec_Cred_Timeout=*value*

Parameter	Description
<i>value</i>	Specifies the number of seconds remaining before a user's network credentials expire and become invalid. You can also specify 'no_limit' (the default) to specify that the credentials not expire. A credential timeout value set by the security system's administrator supersedes any value you specify for Sec_Cred_Timeout.

Default

Sec_Cred_Timeout='no_limit'

Usage

CyberSafe Kerberos If your security mechanism is CyberSafe Kerberos, Sec_Cred_Timeout cannot override the installation default value set for credential timeout.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the *Release* parameter for more information.

Requirements for use To use Sec_Cred_Timeout or any other parameter supporting Open Client security services, you must meet certain requirements for using security services in your InfoMaker application. For details, see "Requirements for using Open Client security services" in *Connecting to Your Database*.

Corresponding CT-Lib connection property Specifying a value for Sec_Cred_Timeout sets the corresponding SAP Sybase CT-Lib connection property named CS_SEC_CREDTIMEOUT.

Examples

To specify 120 seconds (2 minutes) remaining before a user's network credentials expire, type **120** in the Credential Timeout box on the Security page in the Database Profile Setup dialog box.

See also

- Release
- Sec_Channel_Bind
- Sec_Delegation
- Sec_Keytab_File
- Sec_Mechanism
- Sec_Mutual_Auth
- Sec_Network_Auth
- Sec_Server_Principal
- Sec_Sess_Timeout

Sec_Data_Integrity

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, `Sec_Data_Integrity` is one of several parameters that support per-packet security for network-based security services. (For other per-packet security parameters, see the See Also section.)

`Sec_Data_Integrity` controls whether your connection's security mechanism checks the integrity of data transmitted to and from the server. When `Sec_Data_Integrity` is set to 1, the security mechanism analyzes all packets to ensure that their content was not modified during transmission.

When `Sec_Data_Integrity` is set to 0 (the default), no integrity checking is performed.

You must specify a value for `Sec_Data_Integrity` *before* connecting to the database in InfoMaker.

Using third-party security mechanisms

For information about the third-party security mechanisms and operating system platforms that Apeon has tested with Open Client security services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

`Sec_Data_Integrity=value`

Parameter	Description
<i>value</i>	<p>Specifies whether your connection's security mechanism performs integrity checking on data transmitted to and from the server. Values are:</p> <ul style="list-style-type: none"> • 0 (Default) Do <i>not</i> check data integrity. You can also specify 'No' or 'False' to set this value. • 1 Check data integrity by analyzing all packets to ensure that their content was not modified during transmission. You can also specify 'Yes' or 'True' to set this value.

Default

`Sec_Data_Integrity=0`

Usage

When to use Your connection's security mechanism can check data integrity only when your connection is also using network-based login authentication. For information, see your SAP Sybase Open Client/Server documentation.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the [Release](#) parameter for more information.

Requirements for use To use Sec_Data_Integrity or any other parameter supporting Open Client security services, you must meet certain requirements for using security services in your InfoMaker application. For details, see “Requirements for using Open Client security services” in *Connecting to Your Database*.

Corresponding CT-Lib connection property Specifying a value for Sec_Data_Integrity sets the corresponding SAP Sybase CT-Lib connection property named CS_SEC_INTEGRITY.

Examples

To specify that your connection’s security mechanism checks data integrity, select the Ensure Data Integrity check box on the Security page in the Database Profile Setup dialog box.

See also

[Release](#)
[Sec_Confidential](#)
[Sec_Data_Origin](#)
[Sec_Replay_Detection](#)
[Sec_Seq_Detection](#)

Sec_Data_Origin

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, Sec_Data_Origin is one of several parameters that support per-packet security for network-based security services. (For other per-packet security parameters, see the See Also section.)

Sec_Data_Origin controls whether your connection’s security mechanism performs data origin stamping. When Sec_Data_Origin is set to 1, the security mechanism attaches a digital signature to each packet that verifies the packet’s origin and contents.

When Sec_Data_Origin is set to 0 (the default), no data origin stamping is performed.

You must specify a value for Sec_Data_Origin *before* connecting to the database in InfoMaker.

Using third-party security mechanisms

For information about the third-party security mechanisms and operating system platforms that Apeon has tested with Open Client security services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

`Sec_Data_Origin=value`

Parameter	Description
<i>value</i>	<p>Specifies whether your connection's security mechanism performs data origin stamping. Values are:</p> <ul style="list-style-type: none"> • 0 (Default) Do <i>not</i> perform data origin stamping. You can also specify 'No' or 'False' to set this value. • 1 Perform data origin stamping by attaching a digital signature to each packet that verifies the packet's origin and contents. You can also specify 'Yes' or 'True' to set this value.

Default

`Sec_Data_Origin=0`

Usage

Not supported with CyberSafe Kerberos `Sec_Data_Origin` is *not supported* if your security mechanism is CyberSafe Kerberos.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the [Release](#) parameter for more information.

Requirements for use To use `Sec_Data_Origin` or any other parameter supporting Open Client security services, you must meet certain requirements for using security services in your InfoMaker application. For details, see "Requirements for using Open Client security services" in [Connecting to Your Database](#).

Corresponding CT-Lib connection property Specifying a value for `Sec_Data_Origin` sets the corresponding SAP Sybase CT-Lib connection property named `CS_SEC_DATAORIGIN`.

Examples

To specify that your connection's security mechanism performs data origin stamping, select the Verify Packet Origin check box on the Security page in the Database Profile Setup dialog box.

See also

[Release](#)
[Sec_Confidential](#)
[Sec_Data_Integrity](#)

Sec_Replay_Detection
Sec_Seq_Detection

Sec_Delegation

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, Sec_Delegation is one of several parameters that support login authentication for network-based security services. (For other login authentication parameters, see the See Also section.)

For applications that are using network-based login authentication to connect to an SAP Sybase Open Server gateway, Sec_Delegation controls whether the gateway server is allowed to connect to a remote SQL Server using delegated credentials. When Sec_Delegation is set to 1, the gateway can connect to a remote server using the client's delegated credentials. The remote server must also be using network-based authentication and an identical security mechanism.

When Sec_Delegation is set to 0 (the default), the gateway server cannot connect to a remote server using delegated credentials.

You must specify a value for Sec_Delegation *before* connecting to the database in InfoMaker.

Using third-party security mechanisms

For information about the third-party security mechanisms and operating system platforms that Apeon has tested with Open Client security services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

Sec_Delegation=*value*

Parameter	Description
<i>value</i>	<p>Specifies whether the SAP Sybase Open Server gateway is allowed to connect to a remote SQL Server using the client's delegated credentials. Values are:</p> <ul style="list-style-type: none">• 0 (Default) Prohibit the gateway from connecting to a remote server using delegated credentials. You can also specify 'No' or 'False' to set this value.• 1 Allow the gateway to connect to a remote server using delegated credentials. You can also specify 'Yes' or 'True' to set this value.

Default	Sec_Delegation=0
Usage	<p><i>Not supported with CyberSafe Kerberos</i> Sec_Delegation is <i>not supported</i> if your security mechanism is CyberSafe Kerberos.</p> <p><i>Set Release parameter</i> For this parameter to take effect, you <i>must</i> also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the Release parameter for more information.</p> <p><i>Requirements for use</i> To use Sec_Delegation or any other parameter supporting Open Client security services, you must meet certain requirements for using security services in your InfoMaker application. For details, see “Requirements for using Open Client security services” in Connecting to Your Database.</p> <p><i>Corresponding CT-Lib connection property</i> Specifying a value for Sec_Delegation sets the corresponding SAP Sybase CT-Lib connection property named CS_SEC_DELEGATION.</p>
Examples	To allow the Open Server gateway to connect to a remote server using delegated credentials, select the Use Delegated Credentials check box on the Security page in the Database Profile Setup dialog box.
See also	Release Sec_Channel_Bind Sec_Cred_Timeout Sec_Keytab_File Sec_Mechanism Sec_Mutual_Auth Sec_Network_Auth Sec_Server_Principal Sec_Sess_Timeout

Sec_Keytab_File

Description When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, Sec_Keytab_File is one of several parameters that support login authentication for network-based security services. (For other login authentication parameters, see the See Also section.)

Sec_Keytab_File applies only to connections using Distributed Computing Environment (DCE) Kerberos as their security mechanism and requesting network-based login authentication. For those connections, Sec_Keytab_File specifies the name of the keytab file containing the security key for the DCE user.

You *must* set Sec_Keytab_File if the login ID specified in the database profile or Application is *different* from the user name of the DCE user currently running the application.

You must specify a value for Sec_Keytab_File *before* connecting to the database in InfoMaker.

Using third-party security mechanisms

For information about the third-party security mechanisms and operating system platforms that Apeon has tested with Open Client security services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

Sec_Keytab_File='keytab_filename'

Parameter	Description
<i>keytab_filename</i>	The name of the keytab file containing the security key for the DCE user

Default

None

InfoMaker does not set Sec_Keytab_File or the corresponding SAP Sybase Open Client Client-Library (CT-Lib) connection parameter CS_SEC_KEYTAB if you do not specify a value.

Usage

Supported only with Distributed Computing Environment Only Distributed Computing Environment (DCE) security servers and clients support the use of keytab files. Therefore, Sec_Keytab_File is supported only when your security mechanism is DCE Kerberos.

When to use If you want your application to be able to connect to a server with a different user name (login ID) than the DCE user currently running the application, set Sec_Keytab_File to specify the name of the keytab file containing the security key for the appropriate user. For details, see “Requirements for using Open Client security services” in *Connecting to Your Database*.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the [Release](#) parameter for more information.

Requirements for use To use Sec_Keytab_File or any other parameter supporting Open Client security services, you must meet certain requirements for using security services in your InfoMaker application. For details, see “Requirements for using Open Client security services” in *Connecting to Your Database*.

Corresponding CT-Lib connection property Specifying a value for Sec_Keytab_File sets the corresponding SAP Sybase CT-Lib connection property named CS_SEC_KEYTAB.

Examples

To specify *C:\DCE_KEY* as the name of the DCE keytab file, type the following in the Keytab File box on the Security page in the Database Profile Setup dialog box:

```
C:\DCE_KEY
```

See also

Release
 Sec_Channel_Bind
 Sec_Cred_Timeout
 Sec_Delegation
 Sec_Mechanism
 Sec_Mutual_Auth
 Sec_Network_Auth
 Sec_Server_Principal
 Sec_Sess_Timeout

Sec_Mechanism

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, Sec_Mechanism is one of several parameters that support login authentication for network-based security services. (For other login authentication parameters, see the See Also section.)

When you use Open Client security services, you must specify the name of the security mechanism you want to use in the Open Client/Open Server Configuration utility so that the required drivers can be loaded. The default security mechanism is the one currently specified as active in the Configuration utility.

Sec_Mechanism lets you specify a security mechanism name listed in the Open Client/Open Server Configuration utility *other than* the default (active) mechanism.

You must specify a value for Sec_Mechanism *before* connecting to the database in InfoMaker.

Using third-party security mechanisms

For information about the third-party security mechanisms and operating system platforms that Apeon has tested with Open Client security services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

Sec_Mechanism='mechanism_name'

Parameter	Description
<i>mechanism_name</i>	The security mechanism name you want to use to establish a connection. The security mechanism name is case sensitive. You must specify it <i>exactly as it appears</i> in the Open Client/Open Server Configuration utility.

Default

The default value for Sec_Mechanism is the security mechanism name currently specified as active in the Open Client/Open Server Configuration utility. If there is no security mechanism specified, no security service is used to establish the connection.

Usage

When to use Set Sec_Mechanism to use a security mechanism specified in the Open Client/Open Server Configuration utility *other than* the default (active) security mechanism. For instructions on using the Open Client/Open Server Configuration utility, see your SAP Sybase Open Client/Server configuration guide.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the **Release** parameter for more information.

Requirements for use To use Sec_Mechanism or any other parameter supporting Open Client security services, you must meet certain requirements for using security services in your InfoMaker application. For details, see “Requirements for using Open Client security services” in *Connecting to Your Database*.

Corresponding CT-Lib connection property Specifying a value for Sec_Mechanism sets the corresponding SAP Sybase CT-Lib connection property named CS_SEC_MECHANISM.

Examples

To specify KERBEROS as your security mechanism name, type **KERBEROS** in the Security Mechanism box on the Security page in the Database Profile Setup dialog box.

See also

Release
Sec_Channel_Bind
Sec_Cred_Timeout
Sec_Delegation
Sec_Keytab_File
Sec_Mutual_Auth
Sec_Network_Auth
Sec_Server_Principal
Sec_Sess_Timeout

Sec_Mutual_Auth

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, Sec_Mutual_Auth is one of several parameters that support login authentication for network-based security services. (For other login authentication parameters, see the See Also section.)

Sec_Mutual_Auth controls whether your connection's security mechanism performs mutual authentication. When Sec_Mutual_Auth is set to 1, the server must prove its identity to the client before connecting by sending a credential token containing the server's principal name and proof that this name is authentic.

When Sec_Mutual_Auth is set to 0 (the default), no mutual authentication is performed.

You must specify a value for Sec_Mutual_Auth *before* connecting to the database in InfoMaker.

Using third-party security mechanisms

For information about the third-party security mechanisms and operating system platforms that Appeon has tested with Open Client security services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

Sec_Mutual_Auth=*value*

Parameter	Description
<i>value</i>	<p>Specifies whether your connection's security mechanism performs mutual authentication. Values are:</p> <ul style="list-style-type: none"> • 0 (Default) Does <i>not</i> perform mutual authentication. You can also specify 'No' or 'False' to set this value. • 1 Performs mutual authentication. The server must prove its identity to the client before connecting by sending a credential token containing the server's principal name and proof that this name is authentic. You can also specify 'Yes' or 'True' to set this value.

Default

Sec_Mutual_Auth=0

Usage

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the [Release](#) parameter for more information.

Requirements for use To use Sec_Mutual_Auth or any other parameter supporting Open Client security services, you must meet certain requirements for using security services in your InfoMaker application. For details, see “Requirements for using Open Client security services” in [Connecting to Your Database](#).

Corresponding CT-Lib connection property Specifying a value for Sec_Mutual_Auth sets the corresponding SAP Sybase CT-Lib connection property named CS_SEC_MUTUALAUTH.

Examples

To specify that your connection’s security mechanism performs mutual authentication, select the Mutual Authentication check box on the Security page in the Database Profiles dialog box.

See also

- Release
- Sec_Channel_Bind
- Sec_Cred_Timeout
- Sec_Delegation
- Sec_Keytab_File
- Sec_Mechanism
- Sec_Network_Auth
- Sec_Server_Principal
- Sec_Sess_Timeout

Sec_Network_Auth

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, `Sec_Network_Auth` is one of several parameters that support login authentication for network-based security services. (For other login authentication parameters, see the See Also section.)

`Sec_Network_Auth` controls whether your connection uses network-based login authentication. When `Sec_Network_Auth` is set to 1, your connection uses network-based authentication when connecting to a secure **SQL Server**. **Network-based authentication** means that the security mechanism—not the application—confirms that the specified user name represents the authenticated user running the application.

Since the security mechanism rather than the application authenticates your user name (login ID), you need *not* supply a login password for authentication purposes in the database profile or Application if `Sec_Network_Auth` is set to 1.

When `Sec_Network_Auth` is set to 0 (the default), your connection does not use network-based login authentication to connect to the server. You must specify a value for `Sec_Network_Auth` *before* connecting to the database in InfoMaker.

Using third-party security mechanisms

For information about the third-party security mechanisms and operating system platforms that Apeon has tested with Open Client security services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

`Sec_Network_Auth=value`

Parameter	Description
<i>value</i>	<p>Specifies whether your connection uses network-based login authentication when connecting to a secure SQL Server. Values are:</p> <ul style="list-style-type: none"> • 0 (Default) Does <i>not</i> use network-based login authentication when connecting to the server. You can also specify 'No' or 'False' to set this value. • 1 Uses network-based login authentication when connecting to the server. Since the security mechanism rather than the application authenticates your user name (login ID), you need <i>not</i> supply a login password for authentication purposes in the database profile. You can also specify 'Yes' or 'True' to set this value.

Default

Sec_Network_Auth=0

Usage

When to use Setting Sec_Network_Auth to 1 to enable network-based login authentication provides three important benefits for InfoMaker users, because you do not have to specify a login password in the database profile or Application to authenticate the login ID when Sec_Network_Auth is set to 1:

- **Password not stored in registry file** Since you do not specify a login password, no login password is stored in the Windows registry.
- **Password not transmitted across network** Since you do not specify a login password, no login password is transmitted across the network to Adaptive Server.
- **Same user ID and password for different servers** You can use the same network user ID and password to connect to many different Adaptive Server database servers. You can change your password for the network security mechanism and have this change apply on all servers to which your application connects.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the [Release](#) parameter for more information.

Requirements for use To use Sec_Network_Auth or any other parameter supporting Open Client security services, you must meet certain requirements for using security services in your InfoMaker application. For details, see “Requirements for using Open Client security services” in *Connecting to Your Database*.

Corresponding CT-Lib connection property Specifying a value for Sec_Network_Auth sets the corresponding SAP Sybase CT-Lib connection property named CS_SEC_NETWORKAUTH.

Examples

To specify that your connection uses network-based login authentication when connecting to the server, select the Network Based Authentication check box on the Security page in the Database Profile Setup dialog box.

See also

[Release](#)
[Sec_Channel_Bind](#)
[Sec_Cred_Timeout](#)
[Sec_Delegation](#)
[Sec_Keytab_File](#)
[Sec_Mechanism](#)
[Sec_Mutual_Auth](#)
[Sec_Server_Principal](#)
[Sec_Sess_Timeout](#)

Sec_Replay_Detection

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, `Sec_Replay_Detection` is one of several parameters that support per-packet security for network-based security services. (For other per-packet security parameters, see the See Also section.)

`Sec_Replay_Detection` controls whether your connection's security mechanism can detect and reject unauthorized attempts to capture and replay transmitted data. When `Sec_Replay_Detection` is set to 1, the security mechanism detects and subsequently rejects any unauthorized attempts by third parties to capture packets sent to the server and repeat (replay) the commands in the packets at a later time.

When `Sec_Replay_Detection` is set to 0 (the default), the security mechanism cannot detect unauthorized attempts to capture and replay data.

You must specify a value for `Sec_Replay_Detection` *before* connecting to the database in InfoMaker.

Using third-party security mechanisms

For information about the third-party security mechanisms and operating system platforms that Apeon has tested with Open Client security services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

`Sec_Replay_Detection=value`

Parameter	Description
<i>value</i>	<p>Specifies whether your connection's security mechanism can detect and reject unauthorized attempts to capture and replay transmitted data. Values are:</p> <ul style="list-style-type: none"> • 0 (Default) Prohibits your security mechanism from detecting unauthorized attempts to capture and replay transmitted data. You can also specify 'No' or 'False' to set this value. • 1 Allows your security mechanism to detect and reject unauthorized attempts to capture and replay transmitted data. You can also specify 'Yes' or 'True' to set this value.

Default

`Sec_Replay_Detection=0`

Usage

When to use In a nonsecure network, unauthorized third parties might capture the commands sent to a server in order to repeat (replay) these commands at a later date. For example, if packets are sent from the client to the server in the order P1, P2, P3 and the server receives the packets in the order P1, P3, P2, this is considered an attempt to replay the data. Setting Sec_Replay_Detection to 1 ensures that your security mechanism can detect and subsequently reject all such unauthorized attempts to capture and replay data transmitted over the network.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the [Release](#) parameter for more information.

Requirements for use To use Sec_Replay_Detection or any other parameter supporting Open Client security services, you must meet certain requirements for using security services in your InfoMaker application. For details, see “Requirements for using Open Client security services” in [Connecting to Your Database](#).

Corresponding CT-Lib connection property Specifying a value for Sec_Replay_Detection sets the corresponding SAP Sybase CT-Lib connection property named CS_SEC_DETECTREPLAY.

Examples

To allow your security mechanism to detect and reject unauthorized attempts to capture and replay transmitted data, select the Detect Replayed Commands check box on the Security page in the Database Profile Setup dialog box.

See also

[Release](#)
[Sec_Confidential](#)
[Sec_Data_Integrity](#)
[Sec_Data_Origin](#)
[Sec_Seq_Detection](#)

Sec_Seq_Detection

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, Sec_Seq_Detection is one of several parameters that support per-packet security for network-based security services. (For other per-packet security parameters, see the See Also section.)

Sec_Seq_Detection controls whether your connection's security mechanism can detect and reject transmitted packets that arrive at the server in a different order than was originally sent from the client. When Sec_Seq_Detection is set to 1, the security mechanism detects and rejects packets that arrive at the server out of sequence.

When Sec_Seq_Detection is set to 0 (the default), the security mechanism cannot detect packets that arrive at the server out of sequence.

You must specify a value for Sec_Seq_Detection *before* connecting to the database in InfoMaker.

Using third-party security mechanisms

For information about the third-party security mechanisms and operating system platforms that Apeon has tested with Open Client security services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

Sec_Seq_Detection=*value*

Parameter	Description
<i>value</i>	<p>Specifies whether your connection's security mechanism can detect and reject packets that arrive at the server in a different order than the one in which they were sent from the client. Values are:</p> <ul style="list-style-type: none"> • 0 (Default) Prohibit your security mechanism from detecting packets that arrive at the server out of sequence. You can also specify 'No' or 'False' to set this value. • 1 Allow your security mechanism to detect and reject packets that arrive at the server out of sequence. You can also specify 'Yes' or 'True' to set this value.

Default

Sec_Seq_Detection=0

Usage

When to use When transmitting data over a network, commands sent to a server might arrive out of sequence. For example, if packets are sent from the client to the server in the order P1, P2, P3 and the server receives the packets in the order P1, P3, P2, this is considered an out-of-sequence error.

Setting Sec_Seq_Detection to 1 ensures that your security mechanism can detect and subsequently reject packets that arrive at the server out of sequence.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Library (CT-Lib) behavior. See the description of the [Release](#) parameter for more information.

Requirements for use To use Sec_Seq_Detection or any other parameter supporting Open Client security services, you must meet certain requirements for using security services in your InfoMaker application. For details, see “Requirements for using Open Client security services” in *Connecting to Your Database*.

Corresponding CT-Lib connection property Specifying a value for Sec_Seq_Detection sets the corresponding SAP Sybase CT-Lib connection property named CS_SEC_DETECTSEQ.

Examples

To allow your security mechanism to detect and reject packets that arrive at the server out of sequence, select the Detect Sequence Errors check box on the Security page in the Database Profile Setup dialog box.

See also

[Release](#)
[Sec_Confidential](#)
[Sec_Data_Integrity](#)
[Sec_Data_Origin](#)
[Sec_Replay_Detection](#)

Sec_Server_Principal

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, Sec_Server_Principal is one of several parameters that support login authentication for network-based security services. (For other login authentication parameters, see the See Also section.)

Sec_Server_Principal specifies the principal name of the server that you want to access. The **server principal name** is the name by which your security mechanism identifies each server.

If the server name (specified in the database profile or Application) is *different* from the server principal name for the server you want to access, you *must* set Sec_Server_Principal to the correct server principal name in order to connect.

You must specify a value for Sec_Server_Principal *before* connecting to the database in InfoMaker.

Using third-party security mechanisms

For information about the third-party security mechanisms and operating system platforms that Apeon has tested with Open Client security services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

Sec_Server_Principal='server_principal_name'

Parameter	Description
<i>server_principal_name</i>	Specifies the principal name of the server you want to access

Default

None

If you do not specify a value, the security mechanism uses the server's directory entry name, which is the same as the server name specified in the database profile or Application.

Usage

When to use When you use Open Client security services with InfoMaker, the server's directory entry name (which you specify as the server name in the database profile or Application) might differ from the server principal name. In this case, you *must* set Sec_Server_Principal to the correct server principal name so that the security mechanism can identify the server you want to access.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the [Release](#) parameter for more information.

Requirements for use To use Sec_Server_Principal or any other parameter supporting Open Client security services, you must meet certain requirements for using security services in your InfoMaker application. For details, see "Requirements for using Open Client security services" in [Connecting to Your Database](#).

Corresponding CT-Lib connection property Specifying a value for Sec_Server_Principal sets the corresponding SAP Sybase CT-Lib connection property named CS_SEC_SERVERPRINCIPAL.

Examples

To specify SYS12NT as the principal name of the server you want to access, type **SYS12NT** in the Server Principal Name box on the Security page in the Database Profile Setup dialog box.

See also

[Release](#)
[Sec_Channel_Bind](#)

Sec_Cred_Timeout
Sec_Delegation
Sec_Keytab_File
Sec_Mechanism
Sec_Mutual_Auth
Sec_Network_Auth
Sec_Sess_Timeout

Sec_Sess_Timeout

Description

When you access an SAP Adaptive Server Enterprise database in InfoMaker through Open Client, Sec_Sess_Timeout is one of several parameters that support login authentication for network-based security services. (For other login authentication parameters, see the See Also section.)

Some security mechanisms allow applications to set session timeout values for connections using network-based login authentication. For these connections, Sec_Sess_Timeout specifies the number of seconds remaining before a session expires. The session timeout period begins when the connection is opened.

By default, Sec_Sess_Timeout specifies that there is no session timeout limit; the session does not expire. You must specify a value for Sec_Sess_Timeout *before* connecting to the database in InfoMaker.

Using third-party security mechanisms

For information about the third-party security mechanisms and operating system platforms that Appeon has tested with Open Client security services, see the Open Client documentation.

Controls

ASE, SYC SAP Adaptive Server Enterprise

Syntax

Sec_Sess_Timeout=*value*

Parameter	Description
<i>value</i>	Specifies the number of seconds remaining before a session expires. You can also specify 'no_limit' (the default) to indicate that the session does not expire. A session timeout value set by the security system's administrator supersedes any value you specify for Sec_Sess_Timeout.

Default

Sec_Sess_Timeout='no_limit'

Usage *CyberSafe Kerberos* If your security mechanism is CyberSafe Kerberos, Sec_Sess_Timeout cannot override the installation default value set for session timeout.

Set Release parameter For this parameter to take effect, you *must* also set the Release parameter to 11 or higher to specify that your application should use the appropriate version of SAP Sybase Open Client Client-Library (CT-Lib) behavior. See the description of the *Release* parameter for more information.

Requirements for use To use Sec_Sess_Timeout or any other parameter supporting Open Client security services, you must meet certain requirements for using security services in your InfoMaker application. For details, see “Requirements for using Open Client security services” in *Connecting to Your Database*.

Corresponding CT-Lib connection property Specifying a value for Sec_Sess_Timeout sets the corresponding SAP Sybase CT-Lib connection property named CS_SEC_SESSTIMEOUT.

Examples To specify 14,400 seconds (4 hours) remaining before a session expires, type **14400** in the Session Timeout box on the Security page in the Database Profile Setup dialog box.

See also

- Release
- Sec_Channel_Bind
- Sec_Cred_Timeout
- Sec_Delegation
- Sec_Keytab_File
- Sec_Mechanism
- Sec_Mutual_Auth
- Sec_Network_Auth
- Sec_Server_Principal

ServerName

Description Specifies the server name for Oracle client interface (OCI) pooling.

Controls ORA Oracle 11g/12c

Syntax ServerName =value

Parameter	Description
value	Sets the name of the server to use for OCI pooling

Default None

Usage	Oracle client interface pooling for InfoMaker applications is created when you connect to an Oracle server for the first time. The pooling is identified by the server name and character set that are specified in the ServerName and NLS_Charset parameters, respectively. If two Oracle connections are connected to the same Oracle server but use different character sets, the connections must reside in different connection or session pools. All pooling-related DBParm parameters must be set before the initial database connection.
Examples	To specify the server name for OCI pooling, type the name in the Server text box on the Connection page in the Database Profile Setup dialog box.

ServiceComponents

Description	Specifies the global services the OLE DB interface can use.
	<hr/> When to specify ServiceComponents You must specify the ServiceComponents parameter <i>before</i> connecting to the database. <hr/>
Controls	OLE DB
Syntax	ServiceComponents='servicecomponent_name'
Default	None
Examples	To enable the resource pooling service component, select Resource Pooling from the Service Component Support box on the System page in the Database Profile Setup dialog box.

SessionHomogeneous

Description	Authenticates all sessions in a session pool with the user name and password in effect when the session pool was created.
Controls	ORA Oracle 11g/12c
Syntax	SessionHomogeneous =value

Parameter	Description
<i>value</i>	<p>Specifies restrictions on authentication values for a session pool. Values are:</p> <ul style="list-style-type: none"> • True Sessions in the session pool can be authenticated only with the user name and password of the session pool creator. • False (Default) The sessions in the session pool can use the user name and password of a connection request even if they differ from the user name and password of the session pool creator.

Default False, but this value is meaningless if session pooling is not set.

Usage When set to True, all sessions in the session pool are authenticated with the user name and password in effect when the pool was created. The user name and password in later connection requests are ignored. Proxy sessions cannot be created in homogeneous session mode.

Examples To enable homogeneous session pooling, select Session Pooling from the Pooling Type drop-down list on the Pooling page in the Database Profile Setup dialog box and select the Homogeneous Session Pooling check box on the same page.

See also [Pooling](#)

ShowTempTables

Description Specifies whether temporary tables are displayed when you request a list of tables from the Database painter or **SQL** Select painter.

Controls ASE, SYC SAP Adaptive Server Enterprise

Syntax ShowTempTables=*value*

Parameter	Description
<i>value</i>	<p>Specifies whether temporary tables are displayed when you request a list of tables. Values are:</p> <ul style="list-style-type: none"> • 1 (Default) Display temporary tables. • 0 Do not display temporary tables.

Default ShowTempTables=1

Usage The ShowTempTables database parameter applies only to database connections from within the development environment. When ShowTempTables=1, a request for a list of database tables from within a painter causes both sp_pb100table and sp_pb100temptab to execute. The union of these two result sets is displayed to the user. If ShowTempTables=0, only sp_pb100table is executed. The behavior is the same if the stored procedures are not installed on the ASE server.

Examples To specify that you want to display temporary tables, select the Show Temp Tables check box on the System page in the Database Profile Setup dialog box.

ShowWarnings

Description Specifies whether warning message text can be concatenated to existing error messages.

Controls DIR SAP Sybase DirectConnect

Syntax ShowWarnings=*value*

Parameter	Description
<i>value</i>	Specifies whether warning message text can be concatenated to existing error messages. Values are: <ul style="list-style-type: none">• 0 (Default) Does not allow the concatenation of warning message text to existing error messages.• 1 Allows the concatenation of warning message text to existing error messages.

Default ShowWarnings=0

Usage The ShowWarnings parameter allows the DirectConnect interface to use warning and error processing similar to that formerly available in the InformationConnect DB2 Gateway (MDI) interface. For example, if a single warning message appears on the DIR error queue, the default behavior is to discard warnings. If ShowMessages is set, `sqlca.sqlcode=-1` and `sqlca.sqlerrtext="text_of_warning_message"` are returned to the application.

`sqlca.sqlerrtext` cannot exceed 254 characters. Consequently, if multiple warning messages are returned together, message text might be truncated.

For Access Service connections, must be issued before connecting to ensure its correct operation.

Examples To specify that you want to show warning messages as errors, select the Show Warning Messages as Errors check box on the Syntax page in the Database Profile Setup dialog box.

SPCache

Description Specifies the number of stored procedures for which the driver caches information.

When to specify SPCache

If you want to change the default value for SPCache, you must specify a new value *before* connecting to the database. The value cannot be changed at runtime.

Controls OLE DB
O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12c
SNC SQL Native Client for Microsoft SQL Server

Syntax SPCache=*value*

Parameter	Description
<i>value</i>	A number that specifies how many stored procedures are added to a cache that contains information about each stored procedure's parameters. To turn off caching, specify 0.

Default SPCache=100

Usage By default, the driver retrieves information from the server about a stored procedure's parameters the first time the stored procedure is called and caches that information. The next time the procedure is called, the driver retrieves the information from the cache to improve performance. The information is retrieved based on the stored procedure's name. The name is case sensitive.

If you call two different stored procedures with the same name, you can turn off caching by setting the value of SPCache to 0.

Examples To turn off caching of stored procedure parameter information, specify 0 in the Maximum Procedures to Cache box on the System page in the Database Profile Setup dialog box.

SQLCache

Description

Specifies the number of **SQL** statements that InfoMaker should cache. The default is 0, specifying an empty **SQL** cache.

InfoMaker caches SQL statements generated by a report or form.

Controls

ODBC

Syntax

SQLCache=*value*

Parameter	Description
<i>value</i>	The number of cursors you want to open in a script, plus the number of report-generated SELECT statements with retrieval arguments (default=0).

Default

SQLCache=0

Usage

Maintaining statements in the cache Statements in the **SQL** cache are maintained on a least-recently-used (**LRU**) basis. In other words, if a statement must be removed from the cache to make room for another statement, InfoMaker removes the statement that was least recently executed.

SQLCache and bind variables Caching **SQL** statements that you execute frequently improves their performance. Statements with bind variables are often the most frequently used. In fact, if your DBMS does not support bind variables, caching statements is of limited value.

Setting DisableBind to use cached statements In order to use cached statements, make sure the **DisableBind** parameter is set to 0 (the default). This enables the binding of input variables to **SQL** statements.

For more about using bind variables, see [DisableBind](#).

What happens The first time you execute a **SQL** statement containing bind variables, the DataWindow server does the following in this sequence:

- 1 Parses the statement.
- 2 For **SQL SELECT** statements, calls the appropriate database function to get a description of the result set.
- 3 Allocates memory buffers for the bind variables.
- 4 Binds the allocated memory buffers to the parsed statement.

When you cache this **SQL** statement, the DataWindow server stores the parsed statement, result set description, and memory buffer allocation and binding in the **SQL** cache. The next time you execute this statement, InfoMaker finds it in the cache and avoids the overhead of repeating these steps.

If the DataWindow server finds an exact match for this statement in the **SQL** cache, it simply copies the new values supplied for the bind variables to the preallocated memory buffers and executes the statement. This is much faster than having to process the statement from scratch.

Examples

To set the SQL cache size to 25 statements, type **25** in the Number Of SQL Statements Cached box on the Transaction page in the Database Profile Setup dialog box.

See also

[DisableBind](#)

SQLQualifiers**Description**

Sets the level of qualification for identifiers (table and column names) in **SQL** statements when you connect to a database. This affects behavior in forms and reports.

When InfoMaker **qualifies** a table or column name, it prefixes it with the name of the owner. For example, if a user named Fran owns a table named Sales, the qualified table name is Fran.Sales.

Controls

DIR SAP Sybase DirectConnect

Syntax

SQLQualifiers=value

Parameter	Description
<i>value</i>	<p>Sets the level of qualification for identifiers in SQL statements when you connect to a database. Values are:</p> <ul style="list-style-type: none"> • 0 (Default) Do not qualify identifiers with owner names in SQL statements. • 1 Qualify identifiers with owner names in SQL statements.

Default

SQLQualifiers=0

Usage

When InfoMaker qualifies identifiers If the name of the table owner is the same as the name of the user logged in to the database, InfoMaker does not qualify identifiers with owner names in the **SQL** statements it generates. If you set the SQLQualifiers parameter to 1, InfoMaker qualifies identifiers with an owner name in **SQL** statements.

Examples

To specify that you want InfoMaker to qualify identifiers with owner names in **SQL** statements, select the Qualify Identifiers With Owner Names check box on the Syntax page in the Database Profile Setup dialog box.

StatementCache

Description Specifies whether statement caching is enabled and the maximum number of statements to cache.

Controls O10 Oracle 10g
ORA Oracle 11g/12c

Syntax StatementCache='value'

Parameter	Description
value	Specifies whether statement caching is enabled and the maximum number of statements to cache. Values are: <ul style="list-style-type: none"> • 0 Statement caching is disabled. • n Statement caching is enabled and n statements may be cached, where n is a positive integer.

Default StatementCache=0

Usage Statement caching in Oracle provides and manages a cache of statements for each session. On the server, cursors are ready to be used without the need to parse the statement again before execution. Statement caching can be used with connection or session pooling to improve performance and scalability.

Examples To enable statement caching and specify that five statements can be cached, type 5 in the Number of Oracle Statements Cached box on the Transaction page in the Database Profile Setup dialog box.

See also CnnPool
Pooling

StaticBind

Description When you retrieve data from a database into a form or report, InfoMaker does not get a result set description to validate the SELECT statement against the database server before retrieving the data. This means the retrieval should be faster, especially when you are accessing the database over a network. (This feature is called describeless retrieval.)

If you want to override the default behavior and have InfoMaker get a description of the result set before retrieving data, set the StaticBind parameter to 0 or No.

Controls ASE, SYC SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect

JDB JDBC
 ODBC
 OLE DB
 O90 Oracle9i
 O10 Oracle 10g
 ORA Oracle 11g/12c
 SNC SQL Native Client for Microsoft SQL Server

Syntax

StaticBind=*value*

Parameter	Description
<i>value</i>	<p>Specifies whether you want InfoMaker to get a result set description before retrieving data from a database into a form or report. Values are:</p> <ul style="list-style-type: none"> • 0 Get a result set description before retrieving data. You can also specify 'No' to set this value. • 1 (Default) Skip getting a result set description before retrieving data. You can also specify 'Yes' to set this value.

Default

StaticBind=1

Usage

Validation When StaticBind is set to 1 (the default), InfoMaker does not validate the **SELECT** statement against the database server before retrieving data. It assumes that the result set matches the column format of the form or report into which it is being retrieved. If a mismatch occurs, InfoMaker displays an error.

Troubleshooting tips Problems can occur in your application if the result set description obtained by the form or report is different from the current database description of the result set. This can occur for the following reasons:

- The database definition changes after you build the form or report.
- You build the form or report while connected to one DBMS and then run it against a different DBMS.

To fix problems caused by conflicting result set descriptions, you can correct your form or report definition by doing either of the following:

- Export and edit your column definitions
- Force a recompile of the **SQL** statement in the Database painter's Interactive **SQL** (ISQL) view (see the *Users Guide* for instructions)

If your form or report and DBMS result set descriptions do not match and you want to avoid errors, set StaticBind to 0 or No to specify that InfoMaker should *always* get a result set description before retrieving data into a form or report.

Examples

To specify that you want InfoMaker to get a result set description before retrieving data into a form or report, clear the Static Bind check box on the Transaction page (or System page, in the case of the OLE DB interface) in the Database Profile Setup dialog box.

StmtCache

Description

Specifies whether SQL statement caching is enabled on the client.

Controls

I10 Informix

Syntax

StmtCache='value'

Parameter	Description
value	Specifies whether SQL statement caching is enabled on the client. Values are: <ul style="list-style-type: none">• 0 SQL statement caching is disabled on the client.• 1 SQL statement caching is enabled on the client.

Default

StmtCache=0

Usage

In IDS 9.2.1 and later, the database server uses the SQL statement cache (SSC) to store SQL statements across user sessions. When any user executes a statement already stored in the SQL statement cache, the database server does not parse and optimize the statement again, resulting in improved performance. The statement must be a SELECT, UPDATE, DELETE, or INSERT statement, and it cannot contain user-defined routines.

There are several ways to configure caching on the server. The SET STATEMENT CACHE statement takes precedence over the STMT_CACHE environment variable and the STMT_CACHE configuration parameter. You must enable the SQL statement cache, either by setting the STMT_CACHE configuration parameter or by using the Informix onmode utility, before the SET STATEMENT CACHE statement can execute successfully.

You can set the StmtCache DbParameter on the System tab page in the Database Profile Setup dialog box for I10 connections to turn SQL statement caching on or off on the client. However, the server must be configured to support SQL statement caching before you can access the cache from the client.

For more information about Informix SQL statement caching, see the IBM Informix Dynamic Server Performance Guide.

Examples To specify that the client can access data from the **SQL** statement cache if it is configured on the server, select the Enable **SQL** Statement Cache check box on the System page in the Database Profile Setup dialog box.

StrByCharset

Description Specifies how to convert string data between InfoMaker Unicode strings and multibyte strings on the client.

Controls I10 Informix
O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12c

Syntax StrByCharset='value'

Parameter	Description
<i>value</i>	<p>Specifies how to convert string data between InfoMaker Unicode strings and multibyte strings on the client. Values are:</p> <ul style="list-style-type: none"> • 0 String conversion is based on the current OS code page, or, for UTF-8 code sets on Informix, on the UTF-8 code set. • 1 For Informix, string conversion is based on the code set specified in the Client_Locale database parameter. For Oracle, string conversion is based on the Oracle character set specified in the NLS_Charset database parameter.

Default StrByCharset=0

Usage **Informix** The StrByCharset DbParameter specifies how to convert string data between InfoMaker Unicode strings and Informix client multibyte strings. By default, string conversion for UTF-8 code sets is based on the UTF-8 code set, and string conversion for non-UTF-8 code sets is based on the current OS code page. If StrByCharset is set to 1 (true), string conversion is based on the code set specified in the DbParameter Client_Locale.

Example 1 With these settings:

```
StrByCharset=0
Informix Server DB_LOCALE='EN_US.8859-1'
InfoMaker Informix client CLIENT_LOCALE='EN_US.8859-1'
OS code page=1252
```

The Informix client and server character sets match, so all string data from the client can be sent to the server and fetched back directly, even if some characters do not belong to the EN_US.8859-1 character set but are within code page 1252, because no string conversion happens between the Oracle client and server. All string data can be displayed using the OS code page. If StrByCharset is set to 1, when string data is converted between the Informix client (ANSI) and InfoMaker (Unicode), characters that are not in the EN_US.8859-1 character set are not converted correctly.

Example 2 With these settings:

```
StrByCharset=0
Informix Server DB_LOCALE='DE_DE.ASCII'
InfoMaker Informix client CLIENT_LOCALE='DE_DE.ASCII'
OS code page=949 for Korean
```

The client and server character sets match. All string data, including Korean characters, are sent or received one byte at a time so no data is lost. Using the Korean OS code page, all Korean string data can be converted from ANSI to Unicode safely with StrByCharset set to 0.

Example 3 With these settings:

```
StrByCharset=1
Informix Server DB_LOCALE='ZH_TW.BIG5'
InfoMaker Informix client CLIENT_LOCALE='ZH_TW.BIG5'
OS code page=1252
```

Because characters in the ZH_TW.BIG5 character set do not belong to code page 1252, string data conversion must be based on the ZH_TW.BIG5 character set.

Oracle The StrByCharset DbParameter specifies how to convert string data between InfoMaker Unicode and OCI client multibyte strings. By default, string conversion is based on the current OS code page or an Oracle character set. The StrByCharset database parameter is ignored if NLS_Charset is set to Unicode because both InfoMaker and the OCI client use the UTF-16 format. When NLS_Charset is set to another value, you must set StrByCharset to 1 if the character set on the client is incompatible with the OS code page.

Example 1 With these settings:

```
StrByCharset=0
Oracle Server NLS_CHARACTER='WE8ISO8859P1'
```

```
InfoMaker OCI client NLS_CHARSET='WE8ISO8859P1'
OS code page=1252
```

The Oracle client and server character sets match, so all string data from the client can be sent to the server and fetched back directly, even if some characters do not belong to the WE8ISO8859P1 character set but are within code page 1252, because no string conversion happens between the Oracle client and server. All string data can be displayed using the OS code page. If StrByCharset is set to 1, when string data is converted between the OCI client (ANSI) and InfoMaker (Unicode), characters that are not in the WE8ISO8859P1 character set are not converted correctly.

Example 2 With these settings:

```
StrByCharset=0
Oracle Server NLS_CHARACTER='US7ASCII'
InfoMaker OCI client NLS_CHARSET='US7ASCII'
OS code page=949 for Korean
```

The client and server character sets match. All string data, including Korean characters, is sent or received one byte at a time so no data is lost. Using the Korean OS code page, all Korean string data can be converted from ANSI to Unicode safely with StrByCharset set to 0.

Example 3 With these settings:

```
StrByCharset=1
Oracle Server NLS_CHARACTER='ZHT16BIG5'
InfoMaker OCI client NLS_CHARSET='ZHT16BIG5'
OS code page=1252
```

Because characters in the ZHT16BIG5 character set do not belong to code page 1252, string data conversion must be based on the ZHT16BIG5 character set.

Examples

To specify that the Informix client should use the fr_FR.8859-1 character set defined by the Client_Locale parameter to handle string datatypes, specify fr_FR.8859-1 in the Client Locale box and select the String Conversion Based on Client Locale box on the Regional Settings page in the Database Profile Setup dialog box.

To specify that the OCI client should use the character set defined by the NLS_LANG parameter on the local computer to store string datatypes, select Local from the NLS Charset drop-down list and select the Use String Conversion Based on Oracle Character Set box on the System page in the Database Profile Setup dialog box.

See also

[Client_Locale](#)

Db_Locale
NLS_Charset

StripParmNames

Description Specifies that explicitly named parameters should not be passed to the ODBC driver.

Controls ODBC

Syntax StripParmNames=*value*'

Parameter	Description
<i>value</i>	Specifies that explicitly named parameters should not be passed to the ODBC driver. Values are: <ul style="list-style-type: none">• Yes Remove all parameter names from the generated call escape syntax.• No (Default) Keep parameter names that are explicitly specified and include them in the generated call escape syntax.

Default StripParmNames='No'

Usage By default, InfoMaker retains parameter names if explicitly specified in the execution of a stored procedure. As a result, syntax such as the following might be generated and sent to the ODBC driver:

```
{call proc(a=?,b=?)}
```

Some database vendors do not allow parameter names to be specified in the generated call escape syntax. To prevent the passing of explicitly named parameters to the ODBC driver, set StripParmNames to Yes. This means that the parameters are passed in the order specified.

Examples To strip explicitly stated parameter names from a stored procedure, select the Strip Parameter Names check box on the Syntax page in the Database Profile Setup dialog box.

SvrFailover

Description Specifies whether you want InfoMaker to recognize and participate in failover to a designated backup database server if the current database server goes down.

When to specify SvrFailover

You must specify the SvrFailover parameter *before* connecting to the database.

Controls ASE, SYC SAP Adaptive Server Enterprise (12.0 and higher database connections only)
 O90 Oracle9i
 O10 Oracle 10g
 ORA Oracle 11g/12c

Syntax SvrFailover='value'

Parameter	Description
<i>value</i>	<p>Specifies whether you want InfoMaker to recognize and participate in failover to a designated backup database server if the current database server goes down. Values are:</p> <ul style="list-style-type: none"> • No (Default) InfoMaker should not recognize or participate in failover. • Yes InfoMaker should recognize and participate in failover.

Default SvrFailover='No'

Usage Both Oracle and SAP support database server failover in Oracle Version 9.x or later and Adaptive Server 12 or later. For information about how to configure failover for these database servers, see your Oracle or Adaptive Server documentation. To avoid losing your InfoMaker database connection (as the result of a timeout) when a failover takes place, set the SvrFailover parameter so that InfoMaker recognizes and participates in the database server failover.

Oracle also allows you some control over the failover process. Three additional parameters allow you to specify the number of times you want the database server to which you are connected to attempt a failover, how long to wait between failover attempts, and whether InfoMaker should display a runtime dialog box indicating when a failover occurs.

This parameter cannot be set dynamically. The value set when the connection is made remains in effect until it is disconnected.

Examples To tell InfoMaker to recognize and participate in failover, select the Allow Server Failover check box on the Network page in the Database Profile Setup dialog box.

See also FoDelay
FoDialog
FoRetryCount

SystemOwner

Description Specifies the owner of the IBM DB2 system tables that you want InfoMaker to use. InfoMaker accesses the DB2 system tables to get information about the tables and columns in your database.

Controls DIR SAP Sybase DirectConnect

Syntax SystemOwner='owner_name'

Parameter	Description
<i>owner_name</i>	Specifies the owner of the DB2 system tables that you want InfoMaker to use (default=SYSIBM)

Default SystemOwner='SYSIBM'

Usage When you use the SystemOwner parameter to specify a nondefault system owner, InfoMaker uses the set of system tables belonging to this owner instead of the default system tables owned by SYSIBM.

If your site has a large DB2 system catalog, it might be useful to create local copies of the catalog tables and populate them with a subset of the information in the default system catalog. These local copies are sometimes called **shadow catalogs**.

You can then set the value of SystemOwner to the owner of the shadow catalogs. This tells InfoMaker to access the smaller shadow catalogs instead of the larger default system tables, resulting in faster performance. However, you must make sure to keep the shadow catalogs synchronized with the default system catalog owned by SYSIBM.

For more about creating shadow catalogs, see your DB2 system administrator or check whether there is a technical document that describes how to do it. Updated information about connectivity issues is available from [the Appeon Support Web site at https://support.appeon.com/](https://support.appeon.com/).

Examples To specify MYAPP as the owner of the system tables that you want InfoMaker to use, type **MYAPP** in the CSP Catalog Qualifier box on the System page in the Database Profile Setup dialog box.

SystemProcs

Description Specifies whether you want InfoMaker to display both system-stored procedures and user-defined stored procedures in the connected database when you request a list of stored procedures.

By default, InfoMaker displays both system and user-defined stored procedures in the connected database. If you set SystemProcs to 0 or No, only user-defined stored procedures are displayed.

Controls ASE, SYC SAP Adaptive Server Enterprise

Syntax SystemProcs=*value*

Parameter	Description
<i>value</i>	<p>Specifies whether you want InfoMaker to display both system-stored procedures and user-defined stored procedures in the connected database when you request a list of stored procedures. Values are:</p> <ul style="list-style-type: none"> • 0 Display only user-defined stored procedures. You can also specify 'No' to set this value. • 1 (Default) Display both system-stored procedures and user-defined stored procedures. You can also specify 'Yes' to set this value.

Default SystemProcs=1

Usage Setting SystemProcs to 0 or No speeds response time if you want to work only with user-defined stored procedures.

Examples To specify that you want InfoMaker to display only user-defined stored procedures in the connected database when you request a list of stored procedures, clear the Display System Stored Procedures check box on the System page in the Database Profile Setup dialog box.

TableCriteria

Description

Lets you specify search conditions to limit the list of tables and views that displays in the Installed Database Interfaces Tables list in InfoMaker. Setting this parameter can be useful if you are working with a very large database in the InfoMaker development environment.

When to specify TableCriteria

You must specify the TableCriteria parameter *before* connecting to the database.

Controls

ASE, SYC SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect
JDB JDBC
ODBC
O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12c

Syntax

You specify the TableCriteria search conditions on the System page in the Database Profile Setup dialog box.

Default

None. If you do not specify any values, the TableCriteria parameter is not used.

Oracle If you do not specify a value for TableCriteria, all Oracle tables, views, and synonyms that you have permission to access display in the Installed Database Interfaces Tables list by default.

Usage

To specify the TableCriteria search conditions, enter information in the following boxes:

Field	Description
Table Name	<p>Specifies the names of tables to display in the current database. You can use wildcard characters.</p> <p>Default for DirectConnect interface If you omit this value when connected through the DirectConnect interface, InfoMaker displays all tables that you have permission to access in the current database, as defined in the DirectConnect server configuration file.</p> <p>Default for Adaptive Server Enterprise interface If you omit this value when connected through the Adaptive Server Enterprise interface, InfoMaker displays all tables in the current database.</p>

Field	Description
Table Owner	Displays only those tables belonging to the specified table owner. You can use wildcard characters. If you omit this value, InfoMaker displays all tables matching the table name that you have permission to access.
Include Tables	Specifies that tables should be displayed.
Include Views	Specifies that views should be displayed.
Include System Tables	Specifies that system tables should be displayed.

Adaptive Server Enterprise and DirectConnect These SAP database interfaces use stored procedures to create the table list:

- **DirectConnect interface** Uses the `sp_tables` stored procedure.
- **Adaptive Server Enterprise interface** Uses the version of the `sp_pb126table` stored procedure installed by you or your database administrator.

For information about which version of `sp_pb126table` to install when connecting to an Adaptive Server Enterprise database, see “Installing InfoMaker stored procedures in Adaptive Server Enterprise databases” in *Connecting to Your Database*.

InfoMaker uses the `TableCriteria` parameter to supply the arguments to `sp_tables` or `sp_pb126table` and build the table list based on your search criteria.

Examples

Type `QADB%` in the Table Name box and `DWMC31` in the Table Owner box on the System page in the Database Profile Setup dialog box to set the Table Criteria property to:

```
TableCriteria='QADB%,DWMC31'
```

ThreadSafe

Description

Specifies whether your connection should take advantage of the database server threadsafe client libraries.

By default, ThreadSafe is set to No to specify that your connection does not use the threadsafe client libraries. If you set ThreadSafe to Yes, your connection takes advantage of the threadsafe client libraries.

When to specify ThreadSafe

You must specify a value for ThreadSafe *before* connecting to the database.

Controls

I10 Informix
IN9 Informix
O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12c

Syntax

ThreadSafe=*value*

Parameter	Description
<i>value</i>	Specifies whether a connection uses the database server threadsafe client libraries. Values are: <ul style="list-style-type: none">• Yes Your connection uses the threadsafe client libraries. Use this setting when building distributed applications that require a multi-threaded environment.• No (Default) Your connection does not use the threadsafe client libraries. Use this setting when building nondistributed applications that require a single-threaded environment.

Default

ThreadSafe='No'

Usage

When to use Oracle and Informix provide support for thread safety in their client libraries. When you are using the Oracle or Informix database interface to build multi-threaded applications in InfoMaker, set the ThreadSafe parameter to Yes to use threadsafe client libraries. This prevents possible side effects among multiple threads of execution making calls to the database server. Your application might incur a performance penalty when you use the threadsafe client libraries.

By default, the client software (and, thus, InfoMaker) assumes that you are building an application in a single-threaded environment that does not need the threadsafe client libraries. This default ensures that single-threaded applications do not incur the performance penalty associated with using threadsafe libraries. Therefore, if you are building single-threaded applications, you can leave the ThreadSafe parameter set to No (the default).

This parameter cannot be set dynamically. The value set when the connection is made remains in effect until it is disconnected.

Examples

To specify that your connection uses the threadsafe client libraries, select the Thread Safe check box on the Connection page in the Database Profile Setup dialog box.

Time

Description When you update data in the Form painter, InfoMaker builds a **SQL UPDATE** statement in the background. The Time parameter determines how InfoMaker specifies a time datatype when it builds the **SQL UPDATE** statement.

Controls JDB JDBC
 ODBC
 O90 Oracle9i
 O10 Oracle 10g
 ORA Oracle 11g/12c

Syntax The syntax you use to specify the Time parameter differs slightly depending on the database.

The Database Profile Setup dialog box inserts special characters (quotes and backslashes) where needed, so you can specify just the time format.

JDBC and ODBC syntax InfoMaker parses the backslash followed by two single quotes (\'') as a single quote when it builds the **SQL UPDATE** statement:

```
Time=' \'time_format\' '
```

Oracle syntax InfoMaker parses each set of four consecutive single quotes (''''') as a single quote when it builds the **SQL UPDATE** statement:

```
Time=' ''''time_format'''' '
```

Parameter	Description
'\'	JDBC and ODBC syntax Type a single quote, followed by one space, followed by a backslash, followed by two single quotes. There is no space between the two single quotes and the beginning of the time format.
'''''	Oracle syntax Type a single quote, followed by one space, followed by four single quotes. There is no space between the four single quotes and the beginning of the time format.
<i>time_format</i>	The time format you want InfoMaker to use when it builds a SQL UPDATE statement to update a data source in the Form painter. For more on display formats, see the <i>Users Guide</i> .
'\''	JDBC and ODBC syntax Type a backslash, followed by two single quotes, followed by one space, followed by a single quote. There is no space between the end of the time format and the backslash.
'''''	Oracle syntax Type four single quotes, followed by one space, followed by a single quote. There is no space between the end of the time format and the four single quotes.

Default

The default value for Time depends on the DBMS you are accessing:

DBMS	Date default value
JDBC	If no value is specified for the Time parameter, InfoMaker looks for a time format in the section for your JDBC driver in the registry. If no time format is found in the registry, InfoMaker uses the JDBC time format escape sequence.
ODBC	If no value is specified for the Time parameter, InfoMaker looks for a time format in the section for your ODBC driver in the PBODB170 initialization file. If no time format is found in the PBODB170 initialization file, InfoMaker uses the ODBC time format escape sequence.
Oracle	The default Oracle date format. For information, see your Oracle documentation.

Examples

About these examples Assume you are updating a table named Workhours by setting the Start column to 08:30. This time is represented by the following InfoMaker time format:

`hh:mm`

Example 1 (JDBC and ODBC syntax) To specify that InfoMaker should use this format for the time datatype when it builds the SQL UPDATE statement, type `hh:mm` in the Time Format box on the Syntax page in the Database Profile Setup dialog box.

What happens InfoMaker builds the following SQL UPDATE statement to update the table:

```
UPDATE WORKHOURS  
SET START = '08:30'
```

Example 2 (Oracle syntax) To specify that InfoMaker should use this format for the time datatype when it builds the SQL UPDATE statement, type `hh:mm` in the Time Format box on the Syntax page in the Database Profile Setup dialog box.

What happens InfoMaker builds the following SQL UPDATE statement to update the table:

```
UPDATE WORKHOURS  
SET START = '08:30'
```

See also

[Date](#)
[DateTime](#)

TimeFormat

Description When you update data in the Form painter, InfoMaker builds a **SQL UPDATE** statement in the background. The Time parameter determines how InfoMaker specifies a time datatype when it builds the **SQL UPDATE** statement.

Controls OLE DB
SNC SQL Native Client for Microsoft SQL Server

Syntax TimeFormat = '*time_format*'

Parameter	Description
<i>time_format</i>	The time format you want InfoMaker to use when it builds a SQL UPDATE statement to update a data source in the Form painter. For more on display formats, see the <i>Users Guide</i> .

Default If no value is specified for the TimeFormat parameter, InfoMaker does not use a time datatype.

Usage When you call stored procedures, the database server might not accept the time format built by InfoMaker. If this occurs, you can try to use another format. For example, for Microsoft SQL Server, try this format:

```
TimeFormat='\ \'hh:mm:ss\ ''
```

Examples Assume you are updating a table named Workhours by setting the Start column to 08:30. This time is represented by the following InfoMaker time format:

```
hh:mm
```

To specify that InfoMaker should use this format for the time datatype when it builds the SQL UPDATE statement, type `hh:mm` in the Time Format box on the Syntax page in the Database Profile Setup dialog box.

What happens InfoMaker builds the following **SQL UPDATE** statement to update the table:

```
UPDATE WORKHOURS
SET START = '08:30'
```

See also [DateFormat](#)
[DateTimeFormat](#)

Timeout

Description

Specifies the number of seconds the interface should wait for a connection to the data provider to complete.

When to specify TimeOut

You must specify a value for TimeOut *before* connecting to the database.

Controls

OLE DB
SNC SQL Native Client for Microsoft SQL Server

Syntax

TimeOut=*value*

Parameter	Description
<i>value</i>	The number of seconds the interface waits for a connection to complete.

Default

None

Usage

The default value for the TimeOut parameter is driver-specific.

Examples

To set the TimeOut value to wait 10 seconds for a connection to complete, type **10** in the Timeout box on the System page in the Database Profile Setup dialog box.

TimeStamp

Description

Specifies whether InfoMaker should map DateTime and Time datatypes to the Oracle TimeStamp datatype.

Controls

OLE DB
O90 Oracle9i
O10 Oracle 10g
ORA Oracle 11g/12c
SQLNCLI10

Syntax

TimeStamp=*value*

Parameter	Description
<i>value</i>	Specifies whether InfoMaker should map DateTime and Time datatypes to the Oracle TimeStamp datatype. Values are: <ul style="list-style-type: none">• 0 Map DateTime and Time datatypes to the Oracle Date datatype.• 1 (Default) Map DateTime and Time datatypes to the Oracle TimeStamp datatype.

Default	TimeStamp=1
Usage	<p>Oracle9<i>i</i> and later databases and the O90 and O10 interfaces support the Oracle timestamp datatype. This datatype includes the date and the time including milliseconds. The existing Oracle Date datatype does not include millisecond information. In a report, both the Oracle Timestamp and Date datatypes are mapped to the InfoMaker DateTime datatype, which supports millisecond information.</p> <p>If you use the O90 or O10 interface with an Oracle9<i>i</i> or higher server, DateTime and Time datatypes are mapped to the Oracle TimeStamp datatype by default. If you want DateTime and Time to be mapped to the Oracle Date datatype, set the TimeStamp database parameter to 0.</p> <p>In InfoMaker 8 and earlier, millisecond information was truncated when used with the Oracle Date datatype. In InfoMaker 9.0 and later, millisecond information is not truncated. As a result, when performing multiple updates to a DateTime field that maps to a Date column, the first update succeeds, but subsequent updates fail with a “row changed between retrieve and update error.”</p> <p>If you are using an interface with a database that uses only the TimeStamp datatype, InfoMaker handles updates correctly. If the database uses the Oracle Date datatype only, set the DbParameter TimeStamp to 0 to truncate millisecond information.</p>
Examples	To set the TimeStamp value to treat DateTime and Time DataWindow datatypes as Oracle Date datatypes, clear the Map DateTime/Time to Oracle Timestamp check box on the Syntax page in the Database Profile Setup dialog box.

TraceFile

Description	Specifies that the JDBC Driver Manager Trace tool should trace a connection to the database you access through the InfoMaker JDBC interface.
Controls	JDB JDBC
Syntax	TraceFile='tracefile_name'
Default	None

Usage

The JDBC Driver Manager Trace logs errors and informational messages originating from the Driver object currently loaded (such as the SAP Sybase jConnect JDBC driver) when InfoMaker connects to a database through the JDBC interface. It writes this information to a log file (default is JDBC.LOG) or to a file you specify. The amount of trace output varies depending on the JDBC driver being used.

You can start and stop the JDBC Driver Manager Trace in the InfoMaker development environment by editing the database profile for the connection you want to trace.

For more information about using the JDBC Driver Manager Trace tool, see *Connecting to Your Database*.

Examples

To start the JDBC Driver Manager Trace and specify a log file, select the Trace JDBC Calls check box and type the following in the Trace File box on the Options page in the Database Profile Setup dialog box:

```
c:\temp\jdbctrce.log
```

TrimSpaces

Description

Specifies whether InfoMaker should trim trailing spaces from data values retrieved from the following datatypes: Char, Char for Bit Data, VarChar, and VarChar for Bit Data.

Controls

ASE, SYC SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect
I10 Informix
IN9 Informix
JDB JDBC
ODBC
O90 Oracle9i
O10 Oracle 10g
SNC SQL Native Client for Microsoft SQL Server

Syntax

TrimSpaces=*value*

Parameter	Description
<i>value</i>	Specifies whether InfoMaker should trim trailing spaces from data of type Char, Char for Bit Data, and VarChar for Bit Data. Values are: <ul style="list-style-type: none">• 0 Do not trim trailing spaces.• 1 (Default) Trim trailing spaces.

Default TrimSpaces=0 (O90, O10, **SNC** and IN9) or TrimSpaces=1 (DIR, ASE, and SYC). For JDBC, the default values depend on the PBTrimCharColumns value in the registry. For ODBC, the default values depend on the PBTrimCharColumns value in the PBODB nnn .INI file. (If the PBTrimCharColumns keyword is missing for a particular database connection, the default value for the ODBC interface is TrimSpaces=0.)

Usage By default, InfoMaker trims spaces from the following datatypes: Char, Char for Bit Data, VarChar, and VarChar for Bit Data.

ODBC database interface

Some ODBC drivers, such as SQL Anywhere, trim trailing spaces before the data reaches the fetch buffer—even when TrimSpaces is set to 0.

If your DBMS makes a distinction between Char data with trailing spaces and Char data without trailing spaces when evaluating a **WHERE** clause expression, you might receive the message **Row changed between retrieve and update** when your form's update properties are set to "Key and updateable columns." To prevent this, change your form's update properties. In embedded **SQL**, you can check `Sqlca.Sqlnrows` after each update to determine if the update took place. Avoid using Char data columns in the **WHERE** clause of an **UPDATE** or **DELETE** statement when TrimSpaces=1.

Examples To specify that InfoMaker should not trim trailing spaces, clear the Trim Trailing Spaces In CHAR Data check box on the Syntax page in the Database Profile Setup dialog box.

TrustedConnection

Description Specifies whether the current Windows account credentials can be used for authentication.

When to specify TrustedConnection

You must specify the TrustedConnection parameter *before* connecting to the database.

Controls **SNC SQL** Native Client for Microsoft **SQL** Server

Syntax TrustedConnection=*value*

Parameter	Description
<i>value</i>	<p>Specifies whether the current Windows account credentials can be used for authentication. Values are:</p> <ul style="list-style-type: none"> • 0 (Default) The User ID and Password are specified in the connection. • 1 The current Windows account credentials are used for authentication. The User ID and Password supplied in the connection are ignored.

Default

TrustedConnection=0

Examples

To specify that InfoMaker should trust the connection:

- **Database profile** Select the Trusted Connection check box on the General page in the Database Profile Setup dialog box.
- **Application** Type the following in code:

```
SQLCA.DbParameter="TrustedConnection=1"
```

TrustServerCertificate

Description

Specifies whether encryption occurs if there is no verifiable server certificate.

When to specify TrustServerCertificate

You must specify the TrustServerCertificate parameter *before* connecting to the database.

Controls

SNC SQL Native Client for Microsoft SQL Server

Syntax

TrustServerCertificate=*value*

Parameter	Description
<i>value</i>	<p>Specifies whether encryption occurs if there is no verifiable server certificate. The value of this parameter is ignored if the Encrypt DbParameter is not set to 1. Values are:</p> <ul style="list-style-type: none"> • 0 (Default) Encryption occurs only if there is a verifiable server certificate, otherwise the connection attempt fails. • 1 Encryption always occurs, but may use a self-signed server certificate.

Default

TrustServerCertificate=0

Usage SQL Server 2005 always encrypts network packets associated with logging in to the server. If no certificate is provided on the server when it starts up, SQL Server generates a self-signed certificate that is used to encrypt login packets.

SQL Server Configuration Manager can be used to configure the SQL Native Client to request an encrypted connection using the Secure Sockets Layer (SSL), and to accept a self-signed certificate without validation.

You can also request encryption by setting the Encrypt DbParameter to 1, which sets the SQL Native Client connection string keyword Encrypt. To enable encryption to be used when a certificate has not been provided on the server, set both Encrypt and TrustServerCertificate. The value of TrustServerCertificate is ignored if Encrypt is not set.

Examples To specify that InfoMaker should encrypt data and accept the server certificate without validation, select the Encrypt Data and Trust Server Certificate check boxes on the System page in the Database Profile Setup dialog box.

See also [Encrypt](#)

TRS

Description Specifies whether you want your application to connect to a database through the DirectConnect server using:

- DirectConnect Access Service for DB2/MVS
- DirectConnect TRS
- Gatewayless connection using OpenServerConnect

Controls DIR SAP Sybase DirectConnect

Syntax TRS=*value*

Parameter	Description
<i>value</i>	Specifies how your application should connect to a database through the DirectConnect server. Values are: <ul style="list-style-type: none"> • 0 (Default) Use an Access Service connection. • 1 Use a TRS connection. • 2 Use a gatewayless or OpenServerConnect connection.

Default TRS=0

Usage If you have chosen to make a gatewayless database connection, you can then set the UseProcSyntax parameter to specify whether InfoMaker should convert the syntax for invoking a Remote Stored Procedure (RSP) or host-resident request before executing that procedure.

Examples To specify that you want to connect to a database using OpenServerConnect, select Gatewayless from the Choose Gateway drop-down list on the Connection page in the Database Profile Setup dialog box.

See also [UseProcSyntax](#)
[HostReqOwner](#)

UnicharBind

Description Specifies whether PowerBuilder binds string input parameters to the Char or Unichar datatype.

Controls ASE, SYC, SAP Adaptive Server Enterprise

Syntax `UnicharBind=value`

Parameter	Description
<i>value</i>	Specifies whether InfoMaker binds string input parameters to the Char or Unichar datatype. Values are: <ul style="list-style-type: none">• 0 Binds string input parameters as Char datatypes.• 1 Binds string input parameters as Unichar datatypes.

Default UnicharBind=0

Usage The default setting, UnicharBind=0, is recommended for binding Char, Varchar, and Text data. This binding encodes the string data as ANSI strings determined by the current client operating system's code page. UnicharBind=1 is recommended for binding strings as Unichar, Univarchar, and Unitext data.

DisableBind must be set to 0

For UnicharBind to take effect, the DisableBind parameter must be set to 0. DisableBind=1 overrides the UnicharBind setting.

Examples To bind strings to unichar, select the Bind String parameters as Unichar Type check box on the Transaction page in the Database Profile Setup dialog box.

See also [DisableBind](#)
[BindSPInput](#)
[NCharBind](#)
[RPCRebind](#)

URL

Description The location of the database to which you want to connect using the JDBC interface.

When to specify URL

You must specify the URL parameter *before* connecting to the database.

Controls JDB JDBC

Syntax URL='URL_address'

Default None

Usage The database URL is obtained from the database JDBC driver documentation. A list of registered Driver classes, with their URLs, is maintained by the driver's JDBC DriverManager class. When a connection request is made, the DriverManager attempts to locate a suitable driver from those listed.

The URL uses this general format:

jdbc:vendor:driverprotocol:servername:port/database

Argument	Description
<i>jdbc</i>	Driver
<i>vendor</i>	Database vendor (such as SAP or Oracle)
<i>driverprotocol</i>	Database communications protocol
<i>servername</i>	DNS machine name or database host name
<i>port</i>	TCP/IP port number configured for accessing the database server
<i>database</i>	(optional) Name of a specific database

The database URL can also include the user ID and password as follows:

jdbc:vendor:driverprotocol:userid/password@servername:port:database

Examples **Example 1** To set the URL to a database accessed through jConnect, type the following in the URL box on the Connection page in the Database Profile Setup dialog box:

jdbc:sybase:Tds:199.93.178.151:5007/tsdata

Example 2 To set the URL to a database accessed through the Oracle JDBC driver, type the following in the URL box on the Connection page in the Database Profile Setup dialog box:

jdbc:oracle:thin:@ora80nt:1521:orcl

Example 3 To set the URL, which includes the user ID and password, to a database accessed through the Oracle JDBC driver, type the following in the URL box on the Connection page in the Database Profile Setup dialog box:

```
jdbc:oracle:thin:system/manager@ora80nt:1521:orcl
```

See also

Driver

UseProcSyntax

Description

Specifies whether InfoMaker should convert the syntax for invoking a Remote Stored Procedure (RSP) or host-resident request before executing that procedure.

Controls

DIR SAP Sybase DirectConnect (applies only to Open ServerConnect)

Syntax

UseProcSyntax=*value*

Parameter	Description
<i>value</i>	Specifies whether InfoMaker should convert RSP or host-resident request syntax. Values are: <ul style="list-style-type: none">• 0 (Default) Do not convert syntax.• 1 Convert syntax to USE PROCEDURE.

Default

UseProcSyntax=0

Usage

UseProcSyntax applies *only* if you are using Open ServerConnect to make a gatewayless database connection to a DB2/MVS database. To indicate that you want to use Open ServerConnect, select Gatewayless from the Choose Gateway drop-down list on the Connection page. If you select Gatewayless, and set the UseProcSyntax parameter to 1, InfoMaker converts the EXECUTE procedure syntax it normally uses to the USE PROCEDURE syntax required to invoke RSP and host-resident requests.

Examples

To specify that InfoMaker should convert RSP or host-resident syntax, select the Use Procedure Syntax for RSPs check box on the Syntax page in the Database Profile Setup dialog box.

See also

TRS
HostReqOwner

UTF8

Description

The UTF8 database parameter specifies whether the database server you are accessing will handle conversion between the character sets on the client and server when they are different.

When to specify UTF8

You must specify a value for UTF8 *before* connecting to the database in InfoMaker.

Controls

ASE, SYC SAP Adaptive Server Enterprise
DIR SAP Sybase DirectConnect

Syntax

UTF8=*value*

Parameter	Description
<i>value</i>	<p>Specifies whether the database server you are accessing will perform conversion between the character sets used on the client and server. Values are:</p> <ul style="list-style-type: none"> • 0 (Default) The InfoMaker database interface performs the conversion. Multiple languages are not supported. • 1 Your database server performs the conversion. Multiple languages are supported.

Default

UTF8=0

Usage

If UTF8 is set to 1, InfoMaker always uses UTF-8 as the client character set when connecting to an Adaptive Server database. When UTF8 is set to 0, if the client and database server character sets are different, the database interface converts Transact-SQL, identifiers, parameters, and Char and VarChar data to and from the character set used on the server. Multiple languages are not supported with this setting.

To enable multilanguage support, you must set the UTF8 database parameter to 1 and the database server must have the UTF-8 character set installed, or, for Adaptive Server, it must be configured to support Unicode conversions. To do so, the database administrator must run the following command on the server:

```
sp_configure "enable Unicode conversion", 2
```

This enables the server to perform the conversion to and from Unicode.

Examples

To specify that the database server you are accessing with InfoMaker uses UTF-8 as its default character set, select the UTF8 Character Set Installed or Unicode Conversion Enabled check box on the Regional Settings page in the Database Profile Setup dialog box.

About this chapter

This chapter describes the syntax and use of each connection-related database preference that you can set in InfoMaker.

Contents

The database preferences are listed in alphabetical order.

Database preferences and supported database interfaces

The following table lists each supported database interface and the connection-related database preferences you can use with that interface in InfoMaker. The preferences listed in the table pertain to the database connection, and not to the behavior of the Database painter itself.

Database interface	Database preferences
ASE SAP Adaptive Server Enterprise	<ul style="list-style-type: none"> AutoCommit Connect DB at Startup Connect to Default Profile Keep Connection Open Lock Read Only Shared Database Profiles SQL Terminator Character Use Extended Attributes
DIR SAP Sybase DirectConnect	<ul style="list-style-type: none"> AutoCommit Connect DB at Startup Connect to Default Profile Keep Connection Open Lock Read Only Shared Database Profiles SQL Terminator Character Use Extended Attributes

Database interface	Database preferences
Informix	AutoCommit Connect DB at Startup Connect to Default Profile Keep Connection Open Lock Read Only Shared Database Profiles SQL Terminator Character Use Extended Attributes
JDB JDBC	AutoCommit Connect DB at Startup Connect to Default Profile Keep Connection Open Lock Read Only Shared Database Profiles SQL Terminator Character Use Extended Attributes
ODBC <hr/> <p>Using AutoCommit and Lock with ODBC The AutoCommit and Lock database preferences are supported by the ODBC interface only if <i>both</i> the ODBC driver you are using and the back-end DBMS support the feature</p> <hr/>	AutoCommit Connect DB at Startup Connect to Default Profile Keep Connection Open Lock Read Only Shared Database Profiles SQL Terminator Character Use Extended Attributes
OLE DB	AutoCommit Connect DB at Startup Connect to Default Profile Keep Connection Open Lock Read Only Shared Database Profiles SQL Terminator Character Use Extended Attributes
O90 Oracle9i O10 Oracle 10g ORA Oracle 11g/12c	Connect DB at Startup Connect to Default Profile Keep Connection Open Read Only Shared Database Profiles SQL Terminator Character Use Extended Attributes

Database interface	Database preferences
SNC	AutoCommit Connect to Default Profile Keep Connection Open Lock Read Only Shared Database Profiles SQL Terminator Character Use Extended Attributes
SYC SAP Adaptive Server Enterprise	AutoCommit Connect DB at Startup Connect to Default Profile Keep Connection Open Lock Read Only Shared Database Profiles SQL Terminator Character Use Extended Attributes

AutoCommit

Description

For those DBMSs and database interfaces that support it, AutoCommit controls whether InfoMaker issues **SQL** statements outside or inside the scope of a transaction.

When AutoCommit is set to False (the default), InfoMaker issues **SQL** statements *inside* the scope of a transaction. When AutoCommit is set to True, InfoMaker issues **SQL** statements *outside* the scope of a transaction.

When to specify AutoCommit

In the development environment, you must set AutoCommit before connecting to the database. AutoCommit takes effect only when the database connection occurs. Changes to AutoCommit after the connection occurs have no effect on the current connection.

Controls

ADO.NET
 ASE, SYC SAP Adaptive Server Enterprise
 DIR SAP Sybase DirectConnect
 I10 Informix
 IN9 Informix
 JDB JDBC
 ODBC (if driver and back-end DBMS support this feature)

In the development environment	<p>OLE DB SNC SQL Native Client for Microsoft SQL Server</p> <p>Select or clear the AutoCommit Mode check box on the Connection tab in the Database Profile Setup dialog box, as follows:</p> <ul style="list-style-type: none">• Select the check box. Sets AutoCommit to <code>true</code> for this connection.• Clear the check box. (Default) Sets AutoCommit to <code>false</code> for this connection. <p>For instructions, see “Setting Additional Connection Parameters” in <i>Connecting to Your Database</i>.</p>
Default	AutoCommit=False
Usage	<p><i>Transactions</i> A transaction is one or more SQL statements that form a logical unit of work (LUW). Within a transaction, all SQL statements must succeed or fail as one logical entity. Changes are made to the database only if all statements in the transaction succeed and a <code>COMMIT</code> is issued. If one or more statements fail, you must issue a <code>ROLLBACK</code> to undo the changes. This ensures the integrity and security of data in your database.</p> <p><i>Executing SQL DDL statements</i> Some DBMSs require you to execute certain SQL statements outside the scope of a transaction. For example, when connected to a SQL Server 7 or earlier database, you must execute SQL Data Definition Language (DDL) statements such as <code>CREATE TABLE</code> and <code>DROP TABLE</code> outside a transaction. There are two reasons for this:</p> <ul style="list-style-type: none">• It ensures that the structure of your database cannot change during a transaction.• It improves database performance, because DDL statements are costly operations to recover. <p>Therefore, to execute DDL statements or stored procedures containing DDL statements in a SQL Server database, you must set AutoCommit to <code>true</code> to issue the DDL statements outside the scope of a transaction. You should, however, set AutoCommit back to <code>false</code> immediately after executing the DDL statements.</p> <p>When you change the value of AutoCommit from <code>false</code> to <code>true</code>, the DataWindow server issues a <code>COMMIT</code> statement by default.</p>

Caution

When you set AutoCommit to `true`, you cannot roll back database changes. Therefore, use care when changing the setting of AutoCommit.

DirectConnect interface As part of the Connect process, the DIR interface automatically issues `TransactionMode=short` to override the access service default configuration. It then issues `begin transaction` at connect time and after every `Commit` and `Rollback` whenever `AutoCommit=False`. Most developers should start their connections with `AutoCommit=True`, switch to `False` only when the application demands transaction processing, and then switch back to `AutoCommit=True` after the transaction is committed or rolled back.

Examples

To set `AutoCommit` to `true` and issue `SQL` statements outside the scope of a transaction:

- **Development environment** Select the `AutoCommit Mode` check box on the `Connection` tab in the `Database Profile Setup` dialog box.

Connect DB at Startup

Description

When you are working in the InfoMaker development environment, `Connect DB at Startup` controls whether InfoMaker connects to the database when you open a painter requiring a connection (the default) or automatically when you start InfoMaker.

InfoMaker only

The `Connect DB at Startup` preference is available in the `Database Preferences` property sheet only in InfoMaker. It has no effect in InfoMaker applications at runtime or in PowerBuilder®.

Controls

All database interfaces

In the development environment

In the `Database painter`, select or clear the `Connect DB at Startup` check box in the `Database Preferences` property sheet as follows:

- **Select the check box** The next time you start InfoMaker, it automatically connects to the database at startup and stays connected throughout the session until you exit.
- **Clear the check box** (Default) The next time you start InfoMaker, it connects to the database only when you open one of the following painters requiring a connection: `Database`, `Report`, `Form`, `Query`, or `Data Pipeline`. It does *not* connect to the database automatically at startup.

Default

The `Connect DB at Startup` check box in the `Database Preferences` property sheet is cleared by default.

Usage Clearing the Connect DB at Startup check box (the default) in InfoMaker to connect to the database only when a painter requires it can save you money if you are accessing a database that charges for connect time.

Connect to Default Profile

Description Connect to Default Profile controls whether the Database painter establishes a connection to a database using a default profile when the painter is invoked. If not selected, the Database painter opens without establishing a connection to a database.

Controls All database interfaces

In the development environment In the Database painter, select or clear the Connect to Default Profile check box in the Database Preferences dialog box as follows:

- **Select the check box** (Default) The next time you invoke the Database painter, it automatically connects to the default database profile.
- **Clear the check box** The next time you invoke the Database painter, it does *not* automatically connect to the default database profile.

Default The Connect to Default Profile check box in the Database Preferences dialog box is selected by default.

Usage Connect to Default Profile allows you to open the Database painter *without* establishing a connection to a database. Consequently, you can perform all database-related tasks, including defining a database profile and connecting to a database, in the Database painter. However, you might want to continue to define profiles and/or connect to a database using the Database Profile since opening the Database painter uses more system resources.

Keep Connection Open

Description This preference takes effect in the PowerBuilder development environment. The setting of Keep Connection Open has no effect in InfoMaker because the Connect DB at Startup preference controls when InfoMaker connects to the database. (For information, see [Connect DB at Startup on page 199](#).)

Applies to All database interfaces

Default The Keep Connection Open check box in the Database Preferences property sheet is selected by default.

Lock

Description

For those DBMSs and database interfaces that support the use of lock values and isolation levels, the Lock preference sets the isolation level to use when connecting to the database.

In multiuser databases, transactions initiated by different users can overlap. If these transactions access common data in the database, they can overwrite each other or collide.

To prevent concurrent transactions from interfering with each other and compromising the integrity of your database, certain DBMSs allow you to set the isolation level when you connect to the database. **Isolation levels** are defined by your DBMS, and specify the degree to which operations in one transaction are visible to operations in a concurrent transaction. Isolation levels determine how your DBMS isolates or **locks** data from other processes while it is being accessed.

InfoMaker uses the Lock preference to allow you to set various database lock options. Each lock value corresponds to an isolation level defined by your DBMS.

When to specify the Lock value

You must set the Lock value *before* you connect to the database. The Lock value takes effect only when the database connection occurs. Changes to the Lock value after the connection occurs have no effect on the current connection.

Controls

ASE, SYC SAP Adaptive Server Enterprise
 DIR SAP Sybase DirectConnect
 I10 Informix
 IN9 Informix
 JDB JDBC
 ODBC (if driver and back-end DBMS support this feature)
 OLE DB
 SNC SQL Native Client for Microsoft SQL Server

Lock values

The following table lists the lock values and corresponding isolation levels for each database interface that supports locking. You set the lock value in code, and the isolation level in a database profile.

For more about the isolation levels that your DBMS supports, see your DBMS documentation.

Database interface	Lock values	Isolation levels
IN9 and I10 Informix (for OnLine databases only)	Dirty Read Committed Read Cursor Stability Repeatable Read	Dirty Read Committed Read Cursor Stability Repeatable Read
JDB JDBC	RU RC RR TS TN	Read Uncommitted Read Committed Repeatable Read Serializable Transactions Transaction None
ODBC	RU RC RR TS TV	Read Uncommitted Read Committed Repeatable Read Serializable Transactions Transaction Versioning
OLE DB	RU RC RR TS TC	Read Uncommitted Read Committed Repeatable Read Serializable Transactions (default) Chaos
SNC SQL Native Client	RU RC RR SS TS TC	Read Uncommitted Read Committed (default) Repeatable Read Snapshot Serializable Transactions Chaos
SAP Adaptive Server Enterprise	0 1 3	Read Uncommitted Read Committed (default) Serializable Transactions
SAP Sybase DirectConnect	0 1 2 3	Read Uncommitted Read Committed (default) Repeatable Read Serializable Transactions

In the development environment

Select the isolation level you want from the Isolation Level drop-down list on the Connection tab in the Database Profile Setup dialog box.

For instructions, see “Setting Additional Connection Parameters” in *Connecting to Your Database*.

Default

The default lock value depends on how your database is configured. For information, see your DBMS documentation.

Usage

ODBC The TV (Transaction Versioning) setting does *not* apply to **SQL Anywhere** databases.

OLE DB The default value for Lock in the discontinued MSS native interface and the **SNC** interface for Microsoft **SQL Server 2005** is Read Committed, but for OLE DB the default is Serializable Transactions. If you want to connect to **SQL Server 2000** using OLE DB, you can override the default value by specifying a value for Lock in the *PBODB170.INI* file. For example:

```
[Microsoft SQL Server]
...
LOCK='RC'
...
```

The value in the *PBODB170.INI* file is used if you do not change the default in the database profile or set the Lock parameter of the Transaction object in code.

SAP Adaptive Server Enterprise SAP Adaptive Server Enterprise supports the following lock values, which correspond to **SQL Server** isolation levels:

- **0—Read Uncommitted (dirty reads)** Isolation level 0 prevents other transactions from changing data that an uncommitted transaction has already modified (through **SQL** statements such as **UPDATE**).
Other transactions cannot modify the data until the transaction commits, but they can still read the uncommitted data (perform dirty reads). Isolation level 0 prohibits retrieval locks on tables or pages.
Isolation level 0 is valid only for SAP System 10 or higher databases.
- **1—Read Committed** (Default) Isolation level 1 prevents dirty reads by issuing shared locks on tables or pages.

A **dirty read** occurs when one transaction modifies a table row and a second transaction reads that row before the first transaction commits the change. If the first transaction rolls back the change, the information read by the second transaction becomes invalid.

- **3—Serializable Transactions (HOLDLOCK behavior)** Isolation level 3 prevents dirty reads, nonrepeatable reads, and phantoms for the duration of a transaction.

A **nonrepeatable read** occurs when one transaction reads a row and then a second transaction modifies that row. If the second transaction commits the change, subsequent reads by the first transaction produce different results than the original read.

A phantom occurs when one transaction reads a set of rows that satisfy a search condition, and then a second transaction modifies that data through a **SQL INSERT, UPDATE, or DELETE** statement. Subsequent reads by the first transaction using the same search conditions produce a different set of rows than the original read.

Examples

Example 1 To set the Lock value to RC (Read Committed) for a **SQL Anywhere** database:

- **Development environment** Select Read Committed from the Isolation Level drop-down list in the Database Profile Setup dialog box.

Example 2 To set the Lock value to 3 (Serializable Transactions) for an **SAP Adaptive Server Enterprise** database:

- **Development environment** Select Serializable Transactions from the Isolation Level drop-down list in the Database Profile Setup dialog box.

Read Only

Description

Read Only specifies whether InfoMaker should update the extended attribute system tables and any other tables in your database. The extended attribute system tables (also known as the extended catalog) consist of five tables that contain default extended attribute information for your database.

The Read Only setting determines whether you can modify (update) the tables in your database. By default, the Read Only check box is cleared in the Database Preferences dialog box. This means that InfoMaker updates the extended attribute system tables and other tables in your database when you make changes.

If you select the Read Only check box, InfoMaker *does not update* the extended attribute system tables or any other tables in your database. You *cannot* modify (update) information in the extended attribute system tables or any other database tables from the Form or Report painters when the Read Only check box is selected.

Controls	All database interfaces
In the development environment	<p>In the Database painter, select or clear the Read Only check box in the Database Preferences dialog box as follows:</p> <ul style="list-style-type: none"> • Select the check box InfoMaker does not update the extended attribute system tables or any other tables in your database. You <i>cannot</i> modify (update) information in the extended attribute system tables or any other database tables from the Form or Report painters when the Read Only check box is selected. • Clear the check box (Default) InfoMaker updates the extended attribute system tables and any other tables in your database when you modify them.
Default	The Read Only check box in the Database Preferences dialog box is cleared by default.
Usage	<p>If you select the Read Only check box in the Database Preferences dialog box, you cannot modify information in <i>any</i> tables from the Form or Report painters.</p> <ul style="list-style-type: none"> • Therefore, you can use only SELECT and Retrieve statements in the Form or Report painters.
See also	Use Extended Attributes

Shared Database Profiles

Description	<p>Specifies the path name of the InfoMaker initialization file containing the database profiles you want to share.</p> <p>For instructions on sharing database profiles in the InfoMaker development environment, see “Managing Database Connections” in <i>Connecting to Your Database</i>.</p>
Controls	All database interfaces
In the development environment	<p>In the Database painter, supply the path name of the InfoMaker initialization file containing shared profiles in the Shared Database Profiles box in the Database Preferences dialog box. You can type the path name or click the Browse button to display it.</p> <p>For instructions, see “Setting Additional Connection Parameters” in <i>Connecting to Your Database</i>.</p>
Default	The Shared Database Profiles box in the Database Preferences dialog box is blank (unspecified) by default.

Examples

To share database profiles contained in the file *I:\SHARE\IM.INI* on the Windows platform, type or browse to the following in the Shared Database Profiles box in the Database Preferences dialog box:

`I:\SHARE\IM.INI`

SQL Terminator Character

Description

Specifies the **SQL** statement terminator character used by the Database painter's Interactive **SQL (ISQL)** view.

The default terminator character for the **ISQL** view is a semicolon (;). If a semicolon conflicts with the terminator character used by your DBMS syntax, you can change the painter's terminator character by specifying a different character in the **SQL Terminator Character** box in the Database Preferences dialog box. A good choice for a terminator character is the backquote (') character.

Changing the terminator character is recommended when you are using the **ISQL** view to create or execute stored procedures, triggers, and **SQL** scripts.

Controls

All database interfaces

In the development environment

In the Database Preferences dialog box in the Database painter, type the terminator character you want to use in the **SQL Terminator Character** box. For instructions, see "Setting Additional Connection Parameters" in *Connecting to Your Database*.

Default

The default **SQL Terminator Character** value in the Database Preferences dialog box is a semicolon (;).

Usage

The following are typical situations that might require you to change the default **SQL Terminator Character** value:

- **Creating stored procedures and triggers** If you are creating stored procedures and triggers in the **ISQL** view, change the painter's terminator character to one that you do not expect to use in the stored procedure or trigger syntax for your DBMS, such as the backquote (') character.

After you finish using the stored procedure, you can change the terminator character back to a semicolon (;). If you prefer, you can continue to use the new terminator character as long as it does not conflict with any stored procedure or trigger syntax you plan to use.

- **Executing SQL scripts** If you plan to execute any **SQL** scripts in the **ISQL** view, make sure the terminator character used in the script agrees with the terminator character currently set in the view.

Examples To change the SQL statement terminator character in the ISQL view to a backquote (`), type a backquote in the SQL Terminator Character box in the Database Preferences dialog box.

Use Extended Attributes

Description Controls access to the extended attribute system tables by specifying whether you want InfoMaker to create these tables. The extended attribute system tables (also known as the extended catalog) consist of five tables that contain default extended attribute information for your database.

By default, the Use Extended Attributes check box is selected in the Database Preferences dialog box. This setting creates the extended attribute system tables the first time you connect to a database using InfoMaker.

Controls All database interfaces

In the development environment

In the Database painter, select or clear the Use Extended Attributes check box in the Database Preferences dialog box as follows:

- **Select the check box** (Default) Creates the extended attribute system tables when connecting to the database for the first time.
- **Clear the check box** Does *not* create the extended attribute system tables if they do not exist. Instead, the Form and Report painters use the appropriate default values for extended attributes (such as headers, labels, and text color). If the extended attribute system tables already exist, InfoMaker does not use them when you create a new report or form.

Default The Use Extended Attributes check box in the Database Preferences dialog box is selected by default.

Usage If you clear the Use Extended Attributes check box in the Database Preferences dialog box, InfoMaker *does not do* any of the following:

- Create the extended attribute system tables
- Insert, update, or delete rows in the extended attribute system tables
- Select information (such as header names) from the extended attribute system tables
- Execute statements that reference the extended attribute system tables

See also [Read Only](#)

Index

A

- AppDriverName database parameter 7
- AppName database parameter 7
- Async database parameter 8
- asynchronous operations, enabling 8
- authentication information, saving in OLE DB databases 123
- authentication service, specifying in OLE DB databases 90
- AutoCommit database preference 197

B

- backquote (?), as SQL terminator character 206
- bind variables
 - and cached SQL statements 52, 166
 - and default column values 53
 - disabling default binding 51
 - using in SQL statements 52
- BindSPInput database parameter 10
- Block database parameter
 - ODBC 12
 - OLE DB 12
 - Oracle 12
 - SAP Adaptive Server Enterprise 14
 - SAP Sybase DirectConnect 14
- blocking factor, setting for cursors 12, 14

C

- CacheAuthentication database parameter 14
- caching SQL statements
 - with bind variables 52, 166
- CallEscape database parameter 15
- case sensitivity, in Oracle databases 99
- catalogs, DB2 shadow 176
- CDB value, in ODBC connect string 26

- character set, setting 16
- CharSet database parameter 16
- CICS resources, releasing 135
- client locale, setting 17, 45
- Client_Locale database parameter 17, 45
- CnnPool database parameter 18
- columns
 - DateTime data type 41
 - default values and bind variables 53
 - delimiting names 81
 - enclosing names in double quotes 48
 - qualification with DirectConnect interface 167
- CommitOnDisconnect database parameter 19
- concurrency control, optimistic 29
- Connect DB at Startup database preference 199
- connect strings, ODBC 26
- Connect to Default Profile database preference 200
- ConnectAs database parameter 20
- ConnectOption database parameter 21
- ConnectionString database parameter 26
- conventions i
- CSMax database parameter 27
- CSMin database parameter 28
- CursorLib database parameter 28
- CursorLock database parameter 29
- cursors
 - blocking factor 12
 - blocking factor, SAP Sybase Client Library 14
 - keyset-driven, ODBC 30
 - library, ODBC 28
 - locking options, ODBC 29
 - mixed, ODBC 30
 - scrollable, ASE 31
 - scrolling options, INFORMIX interfaces 137
 - scrolling options, ODBC 30
 - setting with ConnectOption database parameter 21
 - update characteristics 32
- CursorScroll database parameter 30
- CursorType database parameter 31
- CursorUpdate database parameter 32

Index

Customer Information Control System (CICS), IBM 135

D

Data Definition Language (DDL) statements, SQL 198

data link file, using in OLE DB 34

Data Pipeline painter

displaying terse error messages 101

inserting rows at one time 88

data source, specifying in OLE DB 35

Database painter, changing SQL terminator character 206

database parameters

and supported database interfaces 1

AppDriverName 7

AppName 7

Async 8

Block, ODBC 12

Block, OLE DB 12

Block, Oracle 12

Block, SAP Adaptive Server Enterprise 14

Block, SAP Sybase DirectConnect 14

CacheAuthentication 14

CallEscape 15

CharSet 16

Client_Locale 17, 45

CnnPool 18

CommitOnDisconnect 19

ConnectAs 20

ConnectOption 21

ConnectString 26

CSMax 27

CSMin 28

CursorLib 28

CursorLock, ODBC 29

CursorScroll, ODBC 30

CursorType 31

CursorUpdate 32

Database 33

DataLink 34

DataSource 35

Date 35

DateFormat 38

DateTime 39

DateTimeAllowed 41

DateTimeFormat 42

DBGetTime 44

DBTextLimit 46, 117

DecimalSeparator 46

DelimitIdentifier 48, 81

DelimitIdentifierToDB 50

DisableBind 51, 88, 166

Driver 54

DS_Alias 55

DS_Copy 57

DS_DitBase 59

DS_Failover 61

DS_Password 63

DS_Principal 65

DS_Provider 66

DS_TimeLimit 68

EncryptPassword 71

FailoverPartner 72

FoDelay 73

FoDialog 74

FoRetryCount 75

FormatArgsAsExp 75

HANotification 77

HomogeneousSession 162

Host 79

HostReqOwner 80

IdentifierQuoteCharacter 81

Identity 82

ImpersonationLevel 84

INET_DBPATH 85

INET_PROTOCOL 85

INET_SERVICE 86

Init_Prompt 87

InsertBlock 88

IntegratedSecurity 90

KeepAlive 90

Language 91

LCID 92

Locale 93

Location 94

Log 95

LoginTimeout 95

LowerCaseIdent 96

MapDateToDateTime 97

MaskPassword 97

MaxConnect 98

MixedCase 99

- Mode 100
- MsgTerse 101
- NumbersInternal 105
- NumericFormat 106
- ObjectMode 108
- OJSyntax 109
- PackageProcs 112
- PacketSize 113
- PacketSize, ODBC 113
- PBCatalogOwner 114
- PBMaxBlobSize 117
- PBTrimCharColumns 119
- PBUseProcOwner 120
- PersistEncrypted 122
- PersistSecurityInfo 123
- PoolCreator 123
- Pooling 124
- PoolPwd 125
- Properties 125
- ProtectionLevel 126
- Provider 127
- ProviderString 128
- PWDialog 118, 130
- PWEncrypt 131
- QualifyPublic 132
- Release 132
- Request 135
- RPCRebind 136
- RTConnBalancing 137
- Scroll 137
- Sec_Channel_Bind 138
- Sec_Confidential 140
- Sec_Cred_Timeout 141
- Sec_Data_Origin 144
- Sec_Delegation 146
- Sec_Keytab_File 147
- Sec_Mechanism 149
- Sec_Mutual_Auth 151
- Sec_Network_Auth 153
- Sec_Replay_Detection 155
- Sec_Seq_Detection 156
- Sec_Server_Principal 158
- Sec_Sess_Timeout 160
- ServiceComponents 162
- ShowTempTables 163
- ShowWarnings 164
- SPCache 165
- SQLCache 166
- SQLCAsServerName 161
- SQLQualifiers 167
- StatementCache 168
- StaticBind 168
- StmtCache 170
- StripParmNames 174
- SvrFailover 175
- SystemOwner 176
- SystemProcs 177
- TableCriteria 178
- ThreadSafe 179
- Time 181, 183
- TimeOut 184
- TraceFile 185
- TrimSpaces 186, 187
- TRS 189
- TrustServerCertificate 69, 188
- URL 191
- UseProcSyntax 192
- UTF8 193
- database preferences
 - and supported database interfaces 195
 - AutoCommit 197
 - Connect to Default Profile 200
 - Lock 201
 - Read Only 204
 - Shared Database Profiles 205
 - SQL Terminator Character 206
 - Use Extended Attributes 207
- Database Profile Setup dialog box
 - AutoCommit Mode check box 197
 - Isolation Level box 201
- database profiles
 - connect string for ODBC data sources 26
 - connection pool 18
 - setting Shared Database Profiles database preference 205
- databases
 - controlling updates 204
 - lock values and isolation levels 201
- DataLink database parameter 34
- DataSource database parameter 35
- Date database parameter 35
- date format 35, 38

Index

DateFormat database parameter 38
DateTime data type, as unique key columns 41
DateTime database parameter 39
DateTime format 39, 42
DateTimeAllowed database parameter 41
DateTimeFormat database parameter 42
DB2 shadow catalogs 176
DB2SYSBP.SQL script, and PBCatalogOwner database 116
DBA, as SQL Anywhere stored procedure owner 121
DBGetTime database parameter 44
DBMS
 database parameters supported for each 1
 database preferences supported for each 195
 lock values and isolation levels 201
DBTextLimit database parameter 46
DDL statements, SQL 198
decimal separators
 setting with DecimalSeparator database parameter 46
 setting with NumericFormat DBParm 107
DecimalSeparator database parameter 46
DelimitIdentifier database parameter 48, 81
DelimitIdentifierToDB database parameter 50
describeless retrieval 168
dirty reads 203
DisableBind database parameter 51, 88, 166
DIT base for SAP Sybase Open Client directory services examples 60
Driver database parameter 54
DS_Alias database parameter 55
DS_Copy database parameter 57
DS_DitBase database parameter 59
DS_Failover database parameter 61
DS_Password database parameter 63
DS_Principal database parameter 65
DS_Provider database parameter 66
DS_TimeLimit database parameter 68
DSN (data source name) value, in ODBC connect strings 26

E

EncryptionPass database parameter 70
EncryptPassword database parameter 71
error messages, displaying terse 101

extended attribute system tables
 controlling creation 207
 controlling updates 204
 table owner, setting 114

F

failover, using in Oracle databases 73, 74
FailoverPartner database parameter 72
FoDelay database parameter 73, 74
FoRetryCount database parameter 75
FormatArgsAsExp database parameter 75

G

GenerateEqualsNull database parameter 77

H

HANotification database parameter 77
Hint database parameter 78
HomogeneousSession database parameter 162
Host database parameter 79
HostReqOwner database parameter 80

I

IBM Informix database interface, see Informix database interface 195
IdentifierQuoteCharacter database parameter 81
Identity database parameter 82
ImpersonationLevel database parameter 84
indexes
 delimiting names 81
 enclosing names in double quotes 48
INET_DBPATH database parameter 85
INET_PROTOCOL database parameter 85
INET_SERVICE database parameter 86
InfoMaker, Connect DB at Startup database preference 199
Informix database interface
 client locale, setting 17, 45

- cursor scrolling options, setting 137
- database parameters 1
- database preferences 195
- decimal separator, setting 46
- INET_DBPATH database parameter 85
- INET_PROTOCOL database parameter 85
- INET_SERVICE database parameter 86
- lock values and isolation levels 201
- ThreadSafe database parameter 179
- Init_Prompt database parameter 87
- InsertBlock database parameter 88
- IntegratedSecurity database parameter 90
- Isolation Level box in Database Profile Setup dialog box 201
- isolation levels and lock values 201

J

- JDBC database interface
 - database parameters 1
 - database preferences 195
 - database URL, setting 191
 - date format 35
 - DateTime format 39
 - driver name, setting 54
 - error messages, displaying terse 101
 - lock values and isolation levels 201
 - login timeout, setting 95
 - numeric format, setting 106
 - properties, setting 97, 125
 - Select Tables list, modifying 178
 - time format 181, 183
 - tracing, setting 185

K

- KeepAlive database parameter 90
- keyset-driven cursors, ODBC 30

L

- Language database parameter 91
- LCID database parameter 92

- Locale database parameter 93
- Location database parameter 94
- Lock database preference 201
- Lock Transaction object property 201
- locking
 - and DBMS isolation levels 201
 - cursors, ODBC 29
 - dirty reads 203
 - nonrepeatable reads 203
 - phantoms 203
- Log database parameter 95
- LOG files, for Adaptive Server 95
- logical unit of work (LUW) 197
- LoginTimeout database parameter 95
- LowerCaseIdent database parameter 96
- LUW 197

M

- MapDateToDateTime database parameter 97
- MaskPassword database parameter 97
- MaxConnect database parameter 98
- Microsoft SQL Native Client database interface
 - obtaining IDENTITY column value 82
- mixed cursors, ODBC 30
- MixedCase database parameter 99
- Mode database parameters 100
- MsgTerse database parameter 101

N

- NCharBind database parameter
 - for Oracle 102
- NCharLiteral database parameter 103
- NLS_Charset database parameter 104
- nonrepeatable reads 203
- NumbersInternal database parameter 105
- NumericFormat database parameter 106

O

- ObjectMode database parameter 108
- ODBC data sources

Index

- connect string, setting 26
- cursor library, setting 28
- cursor locking options, setting 29
- cursor scrolling options, setting 30
- data source name (DSN) in ConnectString database parameter 26
- database parameters 1
- database preferences 195
- date format 35
- DateTime format 39
- error messages, displaying terse 101
- inserting rows in Data Pipeline painter 88
- lock values and isolation levels 201
- network packet size, setting 113
- RPCs, rebinding 136
- time format 181, 183
- ODBC drivers
 - connect string, setting 26
 - cursor library, setting 28
 - cursor locking options, setting 29
 - cursor scrolling options, setting 30
 - database parameters 1
 - database preferences 195
 - error messages, displaying terse 101
 - lock values and isolation levels 201
 - login timeout, setting 95
 - network packet size, setting 113
 - numeric format, setting 106
 - RPCs, rebinding 136
- ODBC interface
 - connect string, setting 26
 - ConnectOption database parameter, using 21
 - cursor blocking factor, setting 12
 - cursor library, setting 28
 - cursor locking options, setting 29
 - cursor scrolling options, setting 30
 - database parameters 1
 - database preferences 195
 - date format 35
 - DateTime format 39
 - decimal separator, setting 46
 - error messages, displaying terse 101
 - inserting rows in Data Pipeline painter 88
 - lock values and isolation levels 201
 - login timeout, setting 95
 - network packet size, setting 113
 - numeric format, setting 106
 - RPCs, rebinding 136
 - Select Tables list, modifying 178
 - stripping parameter names 174
 - time format 181, 183
- OJSyntax database parameter 109
- OLE DB database interface
 - access permission 100
 - authentication service, specifying 90
 - caching authentication information 14
 - cursor blocking factor, setting 12
 - data link file, using 33, 34
 - data protection level, specifying 126
 - data provider, specifying 127
 - data source, specifying 35
 - database parameters 1
 - database preferences 195
 - database server name, identifying 94
 - date format 38
 - DateTime format 42
 - encrypting passwords 71, 73
 - impersonation level 84
 - improving performance 13
 - locale identifier, setting 92
 - lock values and isolation levels 201
 - masking passwords 97
 - maximum blob size, specifying 117
 - prompt, setting 87
 - saving authentication information 123
 - saving encrypted passwords 122
 - ServerCursor database parameter 13
 - SQL Server database, specifying 128
 - timeout, setting 184
 - trimming trailing spaces 119
- OnlineIndex database parameter 111
- optimistic concurrency control 29
- Oracle database interface
 - case sensitivity, setting 99
 - cursor blocking factor, setting 12
 - database parameters 1
 - database preferences 195
 - date format 35
 - DateTime format 39
 - decimal separator, setting 46
 - failover 74, 75, 175
 - listing package objects 112

NumbersInternal database parameter 105
 object mode 108
 PackageProcs database parameter 112
 password expired dialog box, displaying 118,
 130
 QualifyPublic database parameter 132
 Select Tables list, modifying 178
 ThreadSafe database parameter 179
 time format 181, 183
 using Oracle? internal number format 105
 using the public qualifier 132

P

PackageProcs database parameter 112
 packet size, network, setting 113
 parenthesis (right), as SQL terminator character 206
 passwords
 encrypting in OLE DB databases 71
 encrypting in SAP Adaptive Server Enterprise
 databases 131
 in ConnectString database parameter 26
 masking in OLE DB databases 97
 saving encrypted in OLE DB databases 122
 PBCatalogOwner database parameter 114
 PBMaxBlobSize database parameter 117
 PBMaxTextSize database parameter 117
 PBNewSPInvocation database parameter 118
 PBTrimCharColumns database parameter 119
 PBUseProcOwner database parameter 120
 PersistEncrypted database parameter 122
 PersistSecurityInfo database parameter 123
 phantoms 203
 Pooling database parameter 124
 PoolPwd database parameter 125
 Properties database parameter 125
 ProtectionLevel database parameter 126
 Provider database parameter 127
 ProviderString database parameter 128
 PWD (password) value, in ODBC connect string 26
 PWDialog database parameter 130
 PWEncrypt database parameter 131

Q

QualifyPublic database parameter 132

R

Read Only database preference 204
 reads, dirty and nonrepeatable 203
 Release database parameter 132
 Request database parameter 135
 result sets, getting description before retrieval 168
 retrieval arguments, as scientific notation 75
 retrieval, describeless 168
 RetrieveRow event, coding for asynchronous operations
 8
 RPCs, rebinding for ODBC data sources 136
 RTConnBalancing database parameter 137

S

SAP Adaptive Server Enterprise database interface
 application name, setting 7
 character set, setting 16
 cursor blocking factor, setting 14
 database parameters 1
 database preferences 195
 declaring cursors 31, 32
 directory services database parameters 55
 encrypting passwords 131
 failover 175
 language, setting 91
 locale, setting 93
 lock values and isolation levels 201
 logging text and image updates 95
 password expired dialog box, displaying 118, 130
 release, setting 132
 security services database parameters 138
 Select Tables list, modifying 178
 SAP SQL Anywhere 121
 database parameters 1
 SAP Sybase DirectConnect interface
 AutoCommit setting 199
 character set, setting 16
 cursor blocking factor, setting 14
 database parameters 1

- database preferences 195
- declaring cursors 32
- displaying identifier names 96
- language, setting 91
- locale, setting 93
- lock values and isolation levels 201
- resources, releasing 135
- Select Tables list, modifying 178
- setting application name 7
- specifying host request library name 80
- table and column name qualification 167
- trimming trailing spaces 186, 187
- SAP Sybase Open Client directory services
 - DS_Alias database parameter 55
 - DS_Copy database parameter 57
 - DS_DitBase database parameter 59
 - DS_Failover database parameter 61
 - DS_Principal database parameter 63, 65
 - DS_Provider database parameter 66
 - DS_TimeLimit database parameter 68
 - Release database parameter 132
 - third-party directory service providers 66
- SAP Sybase Open Client security services
 - Release database parameter 132
 - Sec_Confidential database parameter 140
 - Sec_Cred_Timeout database parameter 141
 - Sec_Data_Integrity database parameter 143
 - Sec_Data_Origin database parameter 144
 - Sec_Delegation database parameter 146
 - Sec_Keytab_File database parameter 147
 - Sec_Mechanism database parameter 149
 - Sec_Mutual_Auth database parameter 151
 - Sec_Network_Auth database parameter 153
 - Sec_Replay_Detection database parameter 155
 - Sec_Seq_Detection database parameter 156
 - Sec_Server_Principal database parameter 158
 - Sec_Sess_Timeout database parameter 160
- scientific notation for retrieval arguments 75
- scrolling options, cursor
 - INFORMIX interfaces 137
 - ODBC 30
- Sec_Channel_Bind database parameter 138
- Sec_Confidential database parameter 140
- Sec_Cred_Timeout database parameter 141
- Sec_Data_Origin database parameter 144
- Sec_Delegation database parameter 146
- Sec_Keytab_File database parameter 147
- Sec_Mechanism database parameter 149
- Sec_Mutual_Auth database parameter 151
- Sec_Network_Auth database parameter 153
- Sec_Replay_Detection database parameter 155
- Sec_Seq_Detection database parameter 156
- Sec_Server_Principal database parameter 158
- Sec_Sess_Timeout database parameter 160
- security, setting with ConnectOption database parameter 21
- Select Tables list, modifying 178
- semicolon, as default SQL terminator character 206
- ServerCursor database parameter 13
- ServiceComponents database parameters 162
- shadow catalogs, creating in DB2 databases 176
- shared database profiles, setting up 205
- ShowTempTables database parameter 163
- ShowWarnings database parameter 164
- SPCache database parameter 165
- SQL Data Definition Language (DDL) statements 198
- SQL files, DB2SYSPB.SQL 116
- SQL statements
 - allowing DateTime columns as unique key columns 41
 - bind variables 51
 - caching 52
 - issuing inside or outside transactions 197
 - table and column delimiters 81
- SQL terminator character, database preference 206
- SQLCache database parameter 166
- SQLCAsServerName database parameter 161
- SQLQualifiers database parameter 167
- SQLSTATE error prefix, suppressing display 101
- StatementCache database parameter 168
- StaticBind database parameter 168
- StmtCache database parameter 170
- stored procedures
 - Adaptive Server Enterprise, displaying 177
 - ODBC, qualifying with owner name 120
 - Oracle, changing SQL terminator character 206
- StrByCharset database parameter 171
- StripParmNames database parameter 174
- SvrFailover database parameter 175
- Sybase Open Client security services
 - Sec_Channel_Bind database parameter 138
- SYSIBM, prohibited as DB2 table owner 116

system tables, DB2 owner 176
 SystemOwner database parameter 176
 SystemProcs database parameter 177

T

TableCriteria database parameter 178
 tables
 controlling updates 204
 DB2 system owner 176
 delimiting names 81
 enclosing names in double quotes 48
 qualification with DirectConnect interface 167
 Select Tables list, modifying 178
 ThreadSafe database parameter 179
 Time database parameter 181, 183
 TimeOut database parameter 184
 TimeStamp database parameter 184
 TraceFile database parameter 185
 Transaction Router Service 189
 transactions
 issuing SQL statements inside or outside 197
 locking and isolation levels 201
 log, Adaptive Server 95
 releasing resources 135
 TrimSpaces database parameter 186
 TRS database parameter 189
 TrustedConnection database parameter 187
 TrustServerCertificate 69, 188
 typographical conventions i

U

Unicode
 Adaptive Server Enterprise 17, 92, 94, 193
 updating databases, controlling 204
 URL, using in JDBC 191
 Use Extended Attributes database preference 207
 UseProcSyntax database parameter 192
 user IDs, in ConnectString database parameter 26
 UTF8 database parameter 193

