New Features InfoMaker[®] 11.5

Document ID: DC00358-01-1150-01

Last revised: September 2008

Торіс	Page
Display enhancements for reports	1
3D Graph styles	2
RichText Edit style for report columns	3
New report properties	4
Database interface enhancements	4
Native driver support for Oracle 11g	4
Native driver support for MS SQL Server 2008	11
PNG support	19
FDCC compliance	20
Tooltips for report columns and controls	21
Silent install and uninstall	22

Display enhancements for reports

In InfoMaker 11.5, you can enhance the appearance of reports with gradients, transparency settings, and pictures. New properties, including Brushmode and Picture, allow you to change the appearance of the background for a report. Using these properties gives your applications a modern look and feel. You can use the Brushmode property to create background gradient effects that give the report more visual interest.

Copyright 2008 by Sybase, Inc. All rights reserved. Sybase trademarks can be viewed at the Sybase trademarks page

at http://www.sybase.com/detail?id=1011207. Sybase and the marks listed are trademarks of Sybase, Inc. (1) indicates registration in the United States of America. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. 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.

Gradients display transitions from one color to another. Using the Brushmode property, you can choose from several different styles of gradient effects (horizontal, vertical, angled, radial) and then set other properties to customize the appearance of the gradient and transitions. If you choose to use a picture, then you could use Picture property settings to determine the appearance of the picture—whether you want to keep it in its original size, stretch it to fit the report, or tile it in any way inside the report. You can adjust the transparency settings for colors and pictures.

For more information on the new report display properties, see the online Help for the following items:

- Brushmode.property
- Picture.property
- Gradient.property
- Transparency

3D Graph styles

	InfoMaker 11.5 adds 3D rendering to display the 3D graphs (Pie3D, Bar3D, Column3D, Line3D, and Area3D) with a more sophisticated look. The new graph styles let you use data item translucency instead of the overlays used by original 3D graph styles. However, you can still use the 3D graphs with the original rendering style if you want.
	The new graph rendering style are supported for standalone graph controls and for graph controls in an InfoMaker report.
DirectX runtime	The new 3D rendering depends on the DirectX runtime, which will be installed the first time the user selects the feature. After the Render3D check box is initially selected, the DirectX installer will launch. If the check box is selected for an unsupported style, nothing happens until one of the supported 3D graphs is selected. If you opt out of the installation, the Render 3D property is ignored.
Unsupported	The following properties are not supported in the new 3D graph styles.
properties	 Axis: ShadeBackEdge, MajorTic, MajorGridLine, DropLines, MinorDivisions, MinorTic, MinorGridLine, PrimaryLine, SecondaryLine, OriginLine, Frame
	• Text: Underline and Alignment for axis labels and text.

RichText Edit style for report columns

In InfoMaker 11.5, you can use the RichText edit style to display column data in a rich text format, and to use different fonts and colors in the same data field.

Columns that you format with the RichText edit style require considerably more storage space than columns with plain text edit styles. Therefore you should set a minimum of 1 KB for the column width. Otherwise, you can use the RichText edit style with columns that have large text datatypes.

By default, whenever a column with the RichText edit style is edited in the Preview view or at runtime, a font toolbar displays. The font toolbar disappears when the column loses focus. The toolbar can be moved and remembers its last position. You can modify the RichTextToolbarActivation constant on a control to display the default font toolbar whenever a report containing columns with the RichText edit style has focus—whether or not this type of column is selected. You can also modify the constant so that the font toolbar never appears.

Report property	Datatype	Description
RichTextToolbar Activation	RichTextTool barActivation (enumerated)	Specifies when the default font toolbar appears for a report that has columns with the RichText edit style. Values are:
		RichTextToolbarActivationAlways! RichTextToolbarActivationNever! RichTextToolbarActivationOnEdit! (default)

You can customize the toolbar by taking advantages of the properties, events, and methods introduced to support the RichText edit style. If you want to use a customized font toolbar instead of the default toolbar, you must set the RichTextToolbarActivation property to RichTextToolbarActivationNever!.

A new InfoMaker expression allows you to strip the RTF formatting of a RichText column.

For more information about the new property and expression function, see the online Help for the following items:

- RichTextToolbarActivation
- StripRTF

New report properties

InfoMaker 11.5 allows you to manipulate the presentation of reports and their columns and controls in a number of new ways—through the definition of gradient backgrounds, picture backgrounds, scalable transparency settings, and tooltip properties for columns and controls.

- New report background properties
- Bandname.property (gradient properties)
- Brushmode
- Gradient.property
- Picture.property
- Transparency (reports)

New column and control properties

- Background.property
- Transparency (columns and controls)
- Tooltip.property

For descriptions of these properties, look up the property names in the online Help.

Database interface enhancements

InfoMaker 11.5 includes the following database interface enhancements:

- Native driver support for Oracle 11g
- Native driver support for MS SQL Server 2008

Native driver support for Oracle 11g

The InfoMaker 11.5 setup program optionally installs the "ORA" database driver for Oracle 11*g* connections. This driver also supports session and connection pooling.

For more detailed information, see:

- Support for session and connection pooling
- ORA driver support for other recent Oracle features

- Supported Oracle features not related to the ORA driver
- Database profile dialog box changes for the ORA driver

Support for session and connection pooling

	Oracle client interfac when you connect to identified by the serve parameters SQLCA.S Oracle connections a character sets, the con pools. All pooling-re- database connection.	e (OCI) pooling for InfoMaker applications is created an Oracle server for the first time. The pooling is er name and character set which are passed in the DBParm ServerName and NLS_Charset, respectively. If two re connected to the same Oracle server but use different nnections must reside in different connection or session lated DBParm parameters must be set before the initial	
Session pooling	Session pooling means that the application creates and maintains a group of stateless sessions to the database. These sessions are passed to clients as requested. If no session is available, a new one is created. When the client is done with the session, the client releases it to the pool. With session pooling, the number of sessions in the pool can increase dynamically.		
	Session pooling does not support external authentication using an OS account. If a Login ID is not specified in a database connection using an existing session pool, the Login ID of the session pooling creator is used for the connection.		
Connection pooling	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, this parameter is also supported by the ORA driver that you use with Oracle 11g. However, if the Pooling parameter is used with this driver, the CNNPool parameter is ignored.		
Deciding what type of pooling to use	The following table describes the circumstances under which you should myour pooling selection:		
	Choose	When database sessions are	
	Session pooling	Stateless (reusable by middle tier threads) and the number of back-end server processes can cause database scaling problems.	

	Choose	When database sessions are	
	Connection pooling	Stateful (not reusable by middle tier threads) and the number of back-end server processes can cause database scaling problems. The number of physical connections and back-end server processes is reduced by using connection pooling. Therefore many more database sessions can be utilized for the same back-end server configuration.	
	No pooling	Stateful (not reusable by middle tier threads) and the number of back-end server processes will never be large enough to cause scaling issues for the database. EAServer components and MTS components do not support either type of pooling for Oracle databases.	
Load balancing	The Oracle Real Application Clusters (RAC) database option allows a single database to be hosted in multiple instances on multiple nodes of the database server. This adds high availability and failover capacity to the database.		
	Connect time load balancing Balancing of work requests occur at two different times: connect time and runtime. Connect time load balancing oc when a session is first created by the application. This ensures that sessions are part of the pool are well distributed across RAC instances, and that sess on each of the instances have a chance to execute work.		
	For session pools that session in the pool is multiple instances, we more requests go to the service.	a support services at one instance only, the first available adequate. When the pool supports services that span ork requests need to be distributed across instances so that he instances with greater capacity or that provide better	
Runtime connection load balancing to dim serve the work. Run an Oracle 11.1 or his server using OCI se		load balancing You can also use runtime connection ct work requests to the sessions in a session pool that best me connection load balancing is enabled by default when her client is connected to a release 10.2 or higher Oracle sion pooling.	
	The DBParm parameter, RTConnBalancing, supports the runtime connection load balancing feature. It is available only when the Pooling parameter is set to Session Pooling, and it can be set before connection only. By default, when you select Session Pooling for the pooling type, the RTConnBalancing value is true.		

ORA driver support for other recent Oracle features

Client result cache	The InfoMaker ORA driver supports Oracle Client Cache, however this feature depends on your Oracle Server and Client configuration. You can configure the Oracle Client Cache with an <i>init.ora</i> or <i>sqlnet.ora</i> file. Cached queries are annotated with "/*+ result_cache */" hints to indicate that results are stored in the query result cache. To cache the client result set, you must also enable OCI statement caching from InfoMaker applications with the StatementCache DBParm parameter.
Application driver name	An OCI application can choose its own name and set it as a diagnostic aid. The AppDriverName DPBARM parameter allows you to set your own client driver name for the InfoMaker ORA interface. 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.
Client access through a proxy	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 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.
	Limitation on proxy connection without pooling When using a proxy connection without pooling, you must set the NLS_Charset DBPARM to "Local" or to another non-Unicode character set. If you do not change the "Unicode" default value for this DBPARM, the connection fails because the Oracle Client Interface does not accept a Unicode name string for its proxy client attribute.

Support for XMLType
datatypeThe InfoMaker ORA driver supports the Oracle XMLType datatype that was
introduced with Oracle 9*i*. The XMLType datatype is mapped to the InfoMaker
String datatype. However, you cannot use this datatype:

- In the Where clause of an embedded SQL statement or in a report
- As a parameter of a procedure or function, because InfoMaker binds XMLType as a String datatype but Oracle does not support this usage
- In columns that you select directly in an Oracle cursor statement

For example, if the col1 column has an XML datatype, you cannot select this column directly in an Oracle cursor as the following code attempts to do:

```
CREATE OR REPLACE Function p_Ora_sp_char_11 return
  types.cursortype
AS
l_cursor types.cursorType;
begin
open l_cursor for select col1 from t_Ora_sp_char_11;
return l_cursor;
end;
```

However, you can use the following code in an InfoMaker application to obtain string values for the XMLType column that you select in an Oracle cursor statement:

```
CREATE OR REPLACE Function p_Ora_sp_char_11 return
  types.cursortype
AS
l_cursor types.cursorType;
begin
  open l_cursor for select x.coll.getstringval() from
     t_Ora_sp_char_11 x;
return l_cursor;
end;
```

Supported Oracle features not related to the ORA driver

InfoMaker 11.5 applications can support the Oracle features listed in the following table, although these features are not enabled by DPBARM parameters or the ORA driver.

Supported feature	Description
Database Resident Connection Pooling (DRCP)	Provides a connection pool in the database server for typical Web application usage (where the application acquires a database connection, works on it for a relatively short time, and then releases it). Any application can leverage this pool by specifying: POOLED in the EZ Connect string or (SERVER=POOLED) in the TNS connect string.
	Before using this feature, the Oracle DBA must configure and start the DRCP.
Fault Diagnosability in OCI	This infrastructure helps diagnose problems such as those caused by data or metadata corruption, or by database code bugs. Oracle stores incident numbers and diagnostic information outside of the database in the Automated Diagnostic Repository (ADR).
Oracle Call Interface (OCI) Security Enhancement	Prevents disclosure of the database version string before authentication and posts warnings in the event of unauthorized access or when user actions are audited.

Database profile dialog box changes for the ORA driver

Pooling parameters

The database profile dialog box for an Oracle 11g connection includes a Pooling tab that lets you select the following pooling parameters:

Pooling parameter	Description
Pooling Type	You can select Session Pooling, Connection Pooling, or None (default). Sets the Pooling DBParm.
Runtime Connection Load Balancing	This check box is selected by default. It is ignored when you select Connection Pooling or None for the Pooling Type. Sets the RTConnBalancing DBParm.
Homogeneous Session	This check box is not selected by default and is valid for session pooling only. If selected, all sessions in the pool are authenticated with the PoolCreator and PoolPwd DBParm parameters when these are provided. The user name and password in later connection requests are ignored. Proxy sessions cannot be created in homogeneous session mode. Sets the SessionHomogeneous DBParm.
Minimum Number of Sessions	Integer for the minimum number of database connection sessions; value is 1 by default. Sets the CSMin DBParm. This value is ignored when the SessionHomogeneous DBParm is set to false.

Pooling parameter	Description
Maximum Number of Sessions	Integer for the maximum number of database connection sessions; value is 100 by default. Sets the CSMax DBParm.
Increment	Integer for database connection increments per session; value is 1 by default. Sets the CSIncr DBParm. This value is ignored when the SessionHomogeneous DBParm is set to false.
Pool Creator	User name used to create the connection or session pool when the pool is not already created. Sets the PoolCreator DBParm to a string for the user name prior to the database connection. If you do not provide a value for the PoolCreator DBParm, the Transaction object's LogID and LogPass properties are used to create the pooling.
Password	Password used to create the connection or session pool when the pool is not already created. Sets the PoolPwd DBParm to a string for the password for the pool creator.

Additional parameters for Oracle 11g support

Besides the pooling parameters, other database parameters provide additional support for Oracle 11g database connections. These are described in the following table:

Additional parameters for Oracle 11g support	Database profile setup tab page	Description
Application Driver Name (available for ORA driver only)	System	Allows you to set your own client driver name for diagnostic purposes. Sets the AppDriverName DPBARM.
Number of Oracle Statements Cached (supported by the O10 and ORA drivers)	Transaction	The number of statements (0 by default) you can cache for each session. Must be set to a nonzero value to use with the Oracle Client Cache. Sets the StatementCache DBParm.

Additional parameters for Oracle 11g support	Database profile setup tab page	Description
Connect As (supported by all available Oracle drivers, but allows entry of an end user name for the ORA driver only)	Connection	Editable drop-down list where you can enter an end user if you are using proxy authentication, or select the DEFAULT, SYSOPER, or SYSDBA user names. Sets the ConnectAs DBParm. You cannot use connection or session pooling if you set the ConnectAs DBParm to SYSDBA or SYSOPER.

Native driver support for MS SQL Server 2008

InfoMaker support for connections to SQL Server 2008 databases includes new database parameters as well as support for new SQL Server datatypes. To connect to SQL Server 2008 from InfoMaker, you must install the SNC 10.0 driver.

For more detailed information, see:

- New database parameters
- Support for new datatypes in SQL Server 2008
- T-SQL enhancements
- Unsupported SQL Server 2008 features

New database parameters

Provider parameter 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 InfoMaker 11.5 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.
Failover parameter	The FailoverPartner DBParm parameter allows you to set the name of a mirror server, thereby maintaining database availability if a failover event occurs. You can also set the name of the mirror server on the System tab of the Database Profile Setup dialog box for the SNC interface.
	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
Support for new dat	atypes in SQL Server 2008
Date and time	The following table lists new SOL Server 2008 date and time datatypes and the

Date and time datatypes

The following table lists new SQL Server 2008 date and time datatypes and the PowerScript datatypes that they map to:

SQL Server datatype	PowerScript datatype
DATE	Date
TIME	Time (Supports only up to 6 fractional seconds precision although SQL Server datatype supports up to 7 fractional seconds precision.)

	SQL Server datatype	PowerScript datatype	
	DATETIME2	DateTime (Supports only up to 6 fractional seconds precision although SQL Server datatype supports up to 7 fractional seconds precision.)	
	The SQL Server 2008 DATETIMEOFFSET datatype is not supported in InfoMaker 11.5.		
	Precision settings Whe database, InfoMaker inclu Specifications view of the Seconds Precision" in the painter. These fields allow and DATETIME2 column	en you map to a table column in a SQL Server 2008 ides a column labeled "Dec" in the Column e Report painter, and a text box labeled "Fractional Column (Object Details) view of the Database y you to list the precision that you want for the TIME is.	
	The precision setting is for data in a column, InfoMal fractional seconds, even it	table creation only. When retrieving or updating the ker uses only up to six decimal places precision for f you enter a higher precision value for the column.	
Filestream datatype	The FILESTREAM datat an NTFS file system. Trans search, and back up FILE	ype allows large binary data to be stored directly in nsact-SQL statements can insert, update, query, STREAM data.	
	The SQL Server Database Varbinary(max) datatype. Varbinary(max) datatype filestream data, use the Se respectively. To specify th must include the FILEST definition. For example:	e Engine implements FILESTREAM as a The InfoMaker SNC interface maps the to a BLOB datatype, so to retrieve or update electBlob or UpdateBlob SQL statements, lat a column should store data on the file system, you REAM attribute in the Varbinary(max) column	
	CREATE TABLE FST GuidColl uniqu UNIQUE DEFAULT IntCol2 int, varbinaryCol3	<pre>'est (leidentifier ROWGUIDCOL NOT NULL ' NEWID(), varbinary(max) FILESTREAM);</pre>	
	Do not use PowerScript You can access FILESTR functions directly in InfoM access functions cannot be information about accessi MSDN SQL Server Develo	file access functions with FILESTREAM data EAM data by declaring and using the Win32 API faker applications. However, existing InfoMaker file e used to access FILESTREAM files. For more ng FILESTREAM data using Win32 APIs, see the per Center Web site at http://msdn.microsoft.com/en-	

us/library/bb933877(SQL.100).aspx.

Using CLR datatypes in InfoMaker	The binary values of the .NET Common Language Runtime (CLR) datatypes can be retrieved from a SQL Server database as blobs that you could use in InfoMaker applications to update other columns in the database. If their return values are compatible with InfoMaker datatypes, you can use CLR datatype methods in PowerScript, dynamic SQL, embedded SQL or in reports, because the SQL script is executed on the SQL Server side.
	The CLR datatypes can also be mapped to Strings in PowerScript, but the retrieved data is a hexadecimal string representation of binary data.
	You can use the above methods to work with all datatypes that are implemented as CLR datatypes, such as the HierarchyID datatype, Spatial datatypes, and User-defined types.
HierarchyID datatype	HierarchyID is a variable length, system datatype that can store values representing nodes in a hierarchical tree, such as an organizational structure. A value of this datatype represents a position in the tree hierarchy.
	ISQL Usage You can use HierarchyID columns with CREATE TABLE, SELECT, UPDATE, INSERT, and DELETE statements in the ISQL painter. For example:
	CREATE TABLE Emp (EmpId int NOT NULL, EmpName varchar(20) NOT NULL, EmpNode hierarchyid NULL);
	To insert HierarchyID data, you can use the canonical string representation of HierarchyID or any of the methods associated with the HierarchyID datatype as shown below.
	<pre>INSERT into Emp VALUES (1, 'Scott', hierarchyid::GetRoot()); INSERT into Emp VALUES (2, 'Tom' , '/1/');</pre>
	<pre>DECLARE @Manager hierarchyid SELECT @Manager = hierarchyid::GetRoot() FROM Emp INSERT into Emp VALUES (2, 'Tom', @Manager.GetDescendant(NULL,NULL)); DECLARE @Employee hierarchyid SELECT @Employee = CAST('/1/2/3/4/' AS hierarchyid) INSERT into Emp VALUES (2, 'Jim', @Employee);</pre>

You cannot select the HierarchyID column directly since it has binary data, and the ISQL painter Results view does not display binary columns. However, you can retrieve the HierarchyID data as a string value using the ToString method of HierarchyID. For example:

Select EmpId, EmpName, EmpNode.ToString() from Emp;

You can also use the following methods on HierarchyID columns to retrieve its data: GetAncestor, GetDescendant, GetLevel, GetRoot, IsDescendant, Parse, and Reparent. If one of these methods returns a HierarchyID node, then use ToString to convert the data to a string. For example:

```
Select EmpId, EmpName, EmpNode.GetLevel() from Emp;
Select EmpId, EmpName,
EmpNode.GetAncestor(1).ToString() from Emp;
```

HierarchyID columns can be updated using a String value or a HierarchyID variable:

```
Update Emp Set EmpNode = '/1/2/' where EmpId=4;
Delete from Emp where EmpNode = '/1/2/';
```

PowerScript Usage You can use HierarchyID columns in embedded SQL statements for SELECT, INSERT, UPDATE, and DELETE operations. HierarchyID data can be retrieved either as a String or as a Binary(Blob) datatype using the SelectBlob statement.

When using a String datatype to retrieve HierarchyID data, use the ToString method. Otherwise the data will be a hexadecimal representation of the binary HierarchyID value.

The following example shows how you can use HierarchyID methods in embedded SQL:

```
long id
String hid,name
Select EmpId, EmpName, EmpNode.ToString()
    into :id, :name, :hid
    from Emp where EmpId=3;
Select EmpId, EmpName, EmpNode.GetLevel()
    into :id, :name, :hid
    from Emp where EmpId=3;
Blob b
Selectblob EmpNode into :b from Emp where EmpId =2;
```

Report Usage Reports do not directly support the HierarchyID datatype. But you can convert the HierarchyID to a string using the ToString method or an associated HierarchyID method in the data source SQL. For example:

SELECT EmpId, EmpName, EmpNode.ToString() FROM Emp; SELECT EmpId, EmpName, EmpNode.GetLevel() FROM Emp;

Spatial datatypesMicrosoft SQL Server 2008 supports two spatial datatypes: the geometry
datatype and the geography datatype. In SQL Server, these datatypes are
implemented as .NET Common Language Runtime (CLR) datatypes.

Although the InfoMaker SNC interface does not work with CLR datatypes, you can convert the spatial datatypes into strings (with the ToString function) and use them in PowerScript, in the ISQL painter, in embedded SQL, and in reports. This is similar to the way you use the HierarchyID datatype. The SelectBlob SQL statement also lets you retrieve binary values for these datatypes.

The geography and geometry datatypes support eleven different data objects, or instance types, but only seven of these types are instantiable: Points, LineStrings, Polygons, and the objects in an instantiable GeometryCollection (MultiPoints, MultiLineStrings, and MultiPolygons). You can create and work with these objects in a database, calling methods associated with them, such as STAsText, STArea, STGeometryType, and so on. For example:

CREATE TABLE SpatialTable (id int IDENTITY (1,1),
GeomCol geometry);
INSERT INTO SpatialTable (GeomCol) VALUES (
geometry::STGeomFromText(
'LINESTRING (100 100,20 180,180 180)',0));
<pre>select id, GeomCol.ToString() from SpatialTable;</pre>
<pre>select id, GeomCol.STAsText(),</pre>
GeomCol.STGeometryType(),
GeomCol.STArea() from SpatialTable;

User-defined types (UDTs) are implemented in SQL Server as CLR types and integrated with .NET. Microsoft SQL Server 2008 eliminates the 8 KB limit for UDTs, enabling the size of UDT data to expand dramatically.

Although the InfoMaker SNC interface does not directly support UDT datatypes, you can use the ToString method to retrieve data for UDTs in the same way as for other CLR datatypes such as HierarchyId or the spatial datatypes. However, if a UDT datatype is mapped to a String datatype in PowerScript, UDT binary values will be retrieved as hexadecimal strings. To retrieve or update data in binary form (blob) from a UDT, you can use the SelectBlob or UpdateBlob SQL statements, respectively.

You can use any of the associated methods of UDT or CLR datatypes that return compatible data (such as String, Long, Decimal, and so on) for PowerBuilder® applications.

T-SQL enhancements

MERGE statement	The MERGE Transact-SQL statement performs INSERT, UPDATE, or DELETE operations on a target table or view based on the results of a join with a source table. You can use MERGE statement in the ISQL painter and in PowerScript using dynamic SQL. For example
	<pre>String mySQL mySQL = "MERGE INTO a USING b ON a.keycol = b.keycol " & + "WHEN MATCHED THEN "& + "UPDATE SET col1 = b.col1,col2 = b.col2 " & + "WHEN NOT MATCHED THEN " & + "INSERT (keycol, col1, col2, col3)" & + "VALUES (b.keycol, b.col1, b.col2, b.col3) " & + "WHEN SOURCE NOT MATCHED THEN " & + "DELETE;" EXECUTE IMMEDIATE :Mysql;</pre>
	Using the MERGE statement in ISQL A MERGE statement must be terminated by a semicolon. By default the ISQL painter uses a semicolon as a SQL terminating character, so to use a MERGE statement in ISQL, the terminating character must be changed to a colon (:), a forward slash (/), or some other special character.
Grouping sets	GROUPING SETS is an extension of the GROUP BY clause that lets you define multiple groupings in the same query. GROUPING SETS produce a single result set, making aggregate querying and reporting easier and faster. It is equivalent to a UNION ALL operation for differently grouped rows.
	The GROUPING SETS, ROLLUP, and CUBE operators are added to the GROUP BY clause. A new function, GROUPING_ID, returns more grouping-level information than the existing GROUPING function. (The WITH ROLLUP, WITH CUBE, and ALL syntax is not ISO compliant and is therefore deprecated.)
	The following example uses the GROUPING SETS operator and the GROUPING_ID function:
	<pre>SELECT EmpId, Month, Yr, SUM(Sales) AS Sales FROM Sales GROUP BY GROUPING SETS((EmpId, ROLLUP(Yr, Month))); SELECT COL1, COL2, SUM(COL3) AS TOTAL_VAL, GROUPING(COL1) AS C1, GROUPING(COL2) AS C2, GROUPING_ID(COL1, COL2) AS GRP_ID_VALUE</pre>

	FROM TEST_TBL GROUP BY ROLLUP (COL1, COL2);
	You can use the GROUPING SETS operator in the ISQL painter, in PowerScript (embedded SQL and dynamic SQL) and in reports (syntax mode).
Row constructors	Transact-SQL now allows multiple value inserts within a single INSERT statement. You can use the enhanced INSERT statement in the ISQL painter and in PowerScript (embedded SQL and dynamic SQL). For example:
	INSERT INTO Employees VALUES ('tom', 25, 5), ('jerry', 30, 6), ('bok', 25, 3);
	When including multiple values in a single INSERT statement with host variables, you must set the DisableBind DBParm to 1. If you use literal values as in the above example, you can insert multiple rows in a single INSERT statement regardless of the binding setting.
Compatibility level	In SQL Server 2008, the ALTER DATABASE statement allows you to set the database compatibility level (SQL Server version), replacing the sp_dbcmptlevel procedure. You can use this syntax in the ISQL painter and in PowerScript (dynamic SQL). For example:
	ALTER DATABASE <database_name> SET COMPATIBILITY_LEVEL = {80 90 100} 80 = SQL Server 2000 90 = SQL Server 2005 100 = SQL Server 2008</database_name>
	Compatibility level affects behaviors for the specified database only, not for the entire database server. It provides only partial backward compatibility with earlier versions of SQL Server. You can use the database compatibility level as an interim migration aid to work around differences in the behaviors of different versions of the database.
Table hints	The FORCESEEK table hint overrides the default behavior of the query optimizer. It provides advanced performance tuning options, instructing the query optimizer to use an index seek operation as the only access path to the data in the table or view that is referenced by the query. You can use the FORCESEEK table hint in the ISQL painter, in PowerScript (embedded SQL and dynamic SQL), and in reports (syntax mode).
	For example:
	<pre>Select ProductID, OrderQty from SalesOrderDetail with (FORCESEEK);</pre>

Unsupported SQL Server 2008 features

The InfoMaker SNC interface does not support the User-Defined Table Type (a user-defined type that represents the definition of a table structure) that was introduced in SQL Server 2008.

PNG support

InfoMaker 11.5 supports the Portable Networks Graphics (PNG) file format for the images that you use in InfoMaker reports. PNG images display at design time as they do at runtime. Transparent portions of a PNG image display the background color or color gradient of the control to which they are assigned when the control has a background color or color gradient property.

In a Picture control For picture controls without a background color property, transparent portions of a PNG image display the background color of the parent object or control. For example, the InfoMaker Picture control does not have a background color property. If you assign a transparent PNG image to the Picture control, the transparent portions display the background color of the window or object that contains the Picture control.

In a PictureButton control Unlike the Picture control, the PictureButton control does have a background color property in InfoMaker. If you assign a transparent PNG image to the PictureButton control, it displays with the background color assigned to the PictureButton control. If you want the PNG image in a PictureButton control to blend in with the background of the container object for the PictureButton control, you must set the PictureButton's BackColor property to the background color of its container object.

Covering other controls or images If you display a transparent PNG image in a Picture control on a window, and the Picture control covers part of a ListBox control (or other noncontainer control or object), the transparent portion of the PNG image clips the control or object it covers and displays with the background color of its container object. If the container object, such as a window, includes a background image in addition to a background color, the PNG image displays the background color only and clips any background image that it covers. **Restrictions and exceptions** You can deploy PNG images with standard InfoMaker targets. However, PNG images cannot be selected in reports with the RichText or OLE presentation style. Also, if you export a report as a PDF file using the XSLFOP method, PNG images in the report are not saved in the generated PDF.

PNG images are drawn with GDI+ functions rather than with bitmap functions. Therefore the Map3DColors property for Picture and PictureButton controls is ignored when you select a PNG file for the picture image. PNG images are also kept locked while the image is active. This is in contrast to bitmap images, which are released once the image is loaded.

FDCC compliance

	The Federal Desktop Core Configuration (FDCC) is a security standard mandated by the US Office of Management and Budget (OMB). To meet the FDCC security requirements, InfoMaker 11.5 can be installed only by a system administrator. However, InfoMaker and the applications that you develop with InfoMaker are designed to be run in a standard user context without elevated system administration privileges.
	Although most InfoMaker files install by default to Program Files\Sybase subdirectories, write access to these subdirectories is typically restricted to administrative users. Therefore, to meet the FDCC requirements, all writable files are installed, copied, or created in directories where standard users have write access.
FDCC constraints on certain InfoMaker features	Several InfoMaker features might still require write access to <i>Program</i> <i>Files\Sybase</i> subdirectories, or require the ability to add a system printer. For example, to save a report to a PDF file, you must first copy the <i>PSCRIPT.DLL</i> , <i>PSCRIPT.NTF</i> , and <i>PS5UI.DLL</i> files to the <i>Program Files\Sybase\Shared\</i> <i>PowerBuilder\drivers</i> directory, and you must install the Sybase® Datawindow PS printer. This must be done by an administrator before a standard user can save a report to a PDF file.
Files shared by all users	Writable files that are shared by all users are installed in the <i>C:\Documents and</i> <i>Settings\All Users\Documents\Sybase\InfoMaker 11.5</i> directory on Windows XP and Windows 2003, and on Windows Vista and Windows 2008, in <i>C:\Users\Public\Documents\Sybase\InfoMaker 11.5</i> . This includes the:
	InfoMaker Windows Help and compiled HTML Help files

Files reserved for
Individual usersOther writable files are installed in the default Program Files\Sybase
subdirectories, but are copied to different locations the first time a user starts
InfoMaker. In this way, each InfoMaker user gets a private copy of these files.

The following table lists the files that are copied and updated in the directories of all users who run an instance of InfoMaker. The path variable in the table header (*UserName*) stands for the user name of an InfoMaker user. For Windows XP and 2003, this is under the *C*:*Documents and Settings* directory. For Windows Vista and 2008, this is under the *C*:*Users* directory.

In C:\\UserName\ subdirectory	Files copied or updated
On Windows XP and 2003: Local Settings\Application Data\Sybase\InfoMaker 11.5	• Initialization files (IM.INI, PBLAB115.INI, PBODB115.INI)
On Windows Vista and 2008: AppData\Local\Sybase\InfoMaker 11.5	• License files (IM115.LIC, im115_sysam.properties)
On Windows XP and 2003: My Documents\Sybase\InfoMaker 11.5\Tutorial	• Files for the InfoMaker Getting Started tutorial
On Windows Vista and 2008: <i>Documents\Sybase\InfoMaker</i> 11.5\Tutorial	

The locations of writable InfoMaker files reserved for individual use are set in HKEY_CURRENT_USER registry entries for each InfoMaker user. For example, the location of the *IM.INI* file that is copied to each user's local application data directory is registered under the registry key *HKEY_CURRENT_USER\Sybase\InfoMaker\11.5\InitPath.*

Tooltips for report columns and controls

Tooltips display text when the pointer pauses over a report column or control. This text can be used to explain the purpose of the column or control. To use this feature, select the column or control for which you want to create a tooltip and then select the Tooltip tab in the Properties view. You can use the tab to specify:

- Text for the tooltip
- Title for the tooltip
- Color of the background and text
- Icon for the tooltip
- Delay before the tooltip appears and disappears

Whether the tooltip appears as a rectangle or callout bubble

For more information, see Tooltip.property in the online Help.

Silent install and uninstall

You can install and uninstall InfoMaker 11.5 without displaying messages or windows during the setup or removal process. However, you must accept the Sybase license agreement before you can run the silent install file. You can indicate your acceptance by typing the following line in a DOS command box before you type in the silent install batch file command:

SET AgreeToSybaseLicense=true

If the DOS prompt does not display the main InfoMaker installation directory, you must change to that directory. From the same DOS command box, you can then run the silent install using the following syntax:

silentinstall.bat ["lic=*licPathNameOrServerName*" "dir=*directoryName*" "shr=sharedDirectoryName" "log=*logFileName*"]

Modifying the batch file directly

You can indicate your acceptance of the Sybase license agreement by directly editing the *silentinstall.bat* file. You do this by changing the line in the batch file that reads set AgreeToSybaseLicense=false. You must change this line to read set AgreeToSybaseLicense=true. After you make this change, you can double-click the *silentinstall.bat* file to run the install with default selections rather than running it from a command line.

You can use the DOS command line to provide help for silent install parameters. The help is available by typing Help or a question mark after entering silentinstall.bat on the command line.

All of the parameters for the silent install are optional. If you do not provide a value for the "lic" parameter, the installer searches the files in the installer directory in ascending alphabetic order for a valid license file. It uses the first valid license file it finds to install InfoMaker. If you do not provide the "lic" parameter and there is no valid license file in the installer search path, an evaluation version of InfoMaker is installed.

The parameters can be listed in any order and are not case sensitive. Parameter values are also not case sensitive.

Silent install parameter	Description
licPathNameOrServerName	The full path to a valid license file with an LIC extension, or the name of a server hosting a served license. If a license server requires a port number, you can include the port number after the server name separated by a colon. For example: silentinstall.bat "lic=myServer:1688"
directoryName	Indicates the main installation directory. If you omit this parameter, the main product components install to the <i>sysDriver</i> \Program Files\Sybase\ InfoMaker 11.5 directory, where <i>sysDriver</i> is the main computer drive.
sharedDirectoryName	Indicates the shared directory for InfoMaker. If you omit this parameter, this directory installs to <i>sysDriver</i> \Program Files\Sybase\Shared.
logFileName	Names the log file for the installation. If you omit this parameter, the log file is written to the system <i>Temp</i> directory with the file name <i>silentinstall.log</i> .

The following example uses all four parameters for the silent install command:

```
silentinstall.bat "lic=D:\im115\im115.lic"
"dir=D:\im115" "shr=D:\shared" "log=D:\install.log"
```

You run the standard silent uninstall by double-clicking the *silentuninstall.bat* file or running it from a command line.