# RTL improvements

Information gathered from several sources

Comparing source code changes between XE8 and DXS with Beyond Compare

What's New <http://docwiki.embarcadero.com/RADStudio/Seattle/en/What%27s_New>

Release Notes <http://docwiki.embarcadero.com/RADStudio/Seattle/en/Release_Notes>

Bug fix list for RAD Studio 10 Seattle <http://edn.embarcadero.com/article/44561>

What's new in RAD Studio 10 Seattle Webinar

## Data

### ADO

#### TCustomADODataSet

##### function GetCalcFieldTypes: TFieldTypes; override; support for variant

function TCustomADODataSet.GetCalcFieldTypes: TFieldTypes;

begin

Result := inherited GetCalcFieldTypes + [ftVariant];

end;

##### functionGetFieldData() bug fix for variant arrays

**begin**

**if VarIsArray(Data) then**

**SetLength(Buffer, VarArrayHighBound(Data, 1) + 1);**

TDBBitConverter.UnsafeFromVariant(Data, Buffer);

**end;**

### Cloud

Lots of small updated and improvements. Still opportunities to make the code cleaner. More constants and internal methods would help, less duplicate code.

#### CloudAPI

##### Authentication

###### New methods

/// <summary>Creates SHA256 hash in hex from a string</summary>

/// <param name="HashString">The string to hash</param>

/// <returns>The SHA256 hash in hex</returns>

class function GetHashSHA256Hex( HashString: string): string;

/// <summary>Creates SHA256 hash in hex from a Stream</summary>

/// <param name="content">The stream to hash</param>

/// <returns>The SHA256 hash in hex</returns>

class function GetStreamToHashSHA256Hex(const Content: TStream): string;

###### SignString updated

/// <summary>Signs the given string using the SHA1 algorithm</summary>

**/// <param name="Signkey">The Key to use for signing</param>**

/// <param name="StringToSign">The string to sign</param>

/// <returns>The SHA1 signed bytes of the given string</returns>

function SignString(**const Signkey: TBytes;** const StringToSign: string): TArray<Byte>; override;

##### TCloudHTTP

###### Uses native HTTP/SSL instead of Indy/OpenSSL

###### Add AuthCallback method

###### Peer : IIPHTTP is now Client: THTTPClient

##### CloudAPI Deprecations

###### XMLEscape

###### Encode64

###### Decode64

###### EncodeBytes64

###### DecodeBytes64

##### function TCloudService.IssueMergeRequest updated

// Android does not support MERGE as it is not standard

{$IFDEF ANDROID}

Headers.Values['x-method-override'] := 'PATCH';

Headers.Values['PATCHTYPE'] := 'MERGE';

Result := PrepareRequest('PUT', Headers, QueryParameters, QueryPrefix, URL);

{$ELSE}

Result := PrepareRequest('MERGE', Headers, QueryParameters, QueryPrefix, URL);

{$ENDIF}

##### procedure TCloudHTTP.Merge updated

{$IFDEF ANDROID}

FResponse := FClient.MergeAlternative(AURL, RequestStream);

{$ELSE}

FResponse := FClient.Merge(AURL, RequestStream);

{$ENDIF}

#### AmazonAPI

##### Native HTTP/SSL instead of Indy/OpenSSL

##### Authentication abstraction

##### Connection supports default storage endpoint

##### Additional regions supported

/// <summary>Regions which can be used with Amazon Services.</summary>

TAmazonRegion = (amzrNotSpecified, amzrEUWest1, amzrEUCentral1, amzrUSEast1, amzrUSWest1, amzrUSWest2,

amzrAPSoutheast1, amzrAPSoutheast2, amzrAPNortheast1, amzrSAEast1, amzrUSClassic, amzrEU);

##### Property types for Amazon queue

//The item names must exactly match the available Amazon attribute names, with a 3 letter prefix.

/// <summary>The available property types for an Amazon queue.</summary>

TAmazonQueueAttribute = (aqaAll, aqaApproximateNumberOfMessages, aqaApproximateNumberOfMessagesNotVisible,

aqaVisibilityTimeout, aqaCreatedTimestamp, aqaLastModifiedTimestamp,

aqaPolicy, aqaMaximumMessageSize, aqaMessageRetentionPeriod,

aqaQueueArn, aqaApproximateNumberOfMessagesDelayed, aqaDelaySeconds,

aqaReceiveMessageWaitTimeSeconds, aqaRedrivePolicy);

##### Create a Queue with a specific name

##### Content size is now Int64 instead of Integer

##### TAmazonStorageService added BucketRegion argument to methods

##### Use TNetEncoding.Base64

##### Amazon service version updated to 2012-11-05 from 2009-02-01

##### Buckets

###### Supports custom region

###### Supports custom endpoint

###### Position content stream at 0 in Create

###### Buckets recognize SHA256 content

#### AzureAPI

##### Authentication uses TNetEncoding.Base64. DecodeStringToBytes rather than DecodeBytes64

##### TNetEncoding.Html.Encode is used instead of XMLEscape

##### TAzureTableService

###### version is now 2014-02-14 instead of 2009-09-19

###### Fixed memory leak on QueryParams in QueryEntitiesXML

##### TAzureQueueService

###### Set content type to application/x-www-form-urlencoded

##### TAzureBlobService

###### LeaseDuration added to HandleBlobLease

###### LeaseDuration added to AcquireBlobLease

###### CopyBlobInternal

Bug fix for blob location

Change check for response code to 202 from 201

###### ListBlobs BlobURL is updated to include ServiceEndpoint and ContainerName

#### New Data.Cloud.CloudResStrs.pas

##### Only one value currently

resourcestring

SSharedKeyError = 'The shared key cannot be created. Please check that you have the required OpenSSL libraries.';

### DataSnap

#### Connectors

##### DSProxyObjectiveCiOS

/// <summary> Returns properties specific to an ObjectiveC\_iOS 4.2 implementation.</summary>

TDSClientProxyWriterObjectiveCiOS42 = class(TDSClientProxyWriterObjectiveCiOS)

public

function Properties: TDSProxyWriterProperties; override;

end;

/// <summary> Returns properties specific to an ObjectiveC\_iOS 7.1 implementation.</summary>

TDSClientProxyWriterObjectiveCiOS71 = class(TDSClientProxyWriterObjectiveCiOS)

public

function Properties: TDSProxyWriterProperties; override;

end;

/// <summary> Returns properties specific to an ObjectiveC\_iOS 8.1 implementation.</summary>

TDSClientProxyWriterObjectiveCiOS81 = class(TDSClientProxyWriterObjectiveCiOS)

public

function Properties: TDSProxyWriterProperties; override;

end;

#### DataSnap

##### Data.FireDACJSONReflect

###### Now uses System.NetEncoding and System.ZLib

##### DataSnap.DSClientResStrs

###### Added

sUnsupportedProtocol = 'Unsupported protocol: %s';

sResponseObjectNotFound = 'HTTP Response Object not found or available yet';

##### TDSCustomRestConnection

###### Source code got reformatted

###### New events

OnValidateCertificate

OnAuthentication

OnSelectClientCertificate

###### New methods

procedure ClearSessionCredentials

function GetSessionCredentials : TArray<string>

###### Changed if Assigned(foo) to if foo <> nil, I don't know why

##### DataSnap.DSHTTPClient TDSHTTP

###### Significantly refactored

###### New internal interfaces

IProxyConnectionInfo

IHeaderList

IRequestHeaderList

IRequest

IResponse

##### DataSnap.DSHTTPWebBroker TDSHTTPRequestWebBroker

###### GetPostStream now uses ReadTotalContent, much cleaner

#### DBX

##### Data.DBXDataSets.pas TDBXReaderDataSet.LoadFieldDef

##### **if ValueType.Precision > High(FldDesc.iUnits1) then**

##### **FldDesc.iUnits1 := High(FldDesc.iUnits1)**

##### **else**

##### FldDesc.iUnits1 := ValueType.Precision;

##### **if ValueType.Scale > High(FldDesc.iUnits2) then**

##### **FldDesc.iUnits2 := High(FldDesc.iUnits2)**

##### **else**

##### FldDesc.iUnits2 := ValueType.Scale;

##### Data.DBXJSONReflect.pas

###### TTypeMarshaller<TSerial: class>

FObjectHash: TDictionary<NativeInt, Integer> instead of TDictionary<Integer, Integer>

###### TTypeMarshaller<TSerial>.MarshalValue

Support for TList<T>.FListHelper on TypeKind.tkRecord

###### TJSONUnMarshal.SetFieldArray

Support for TList<T>.FListHelper on TypeKind.tkRecord

##### Data.DBXMedaDataCommandFactory.pas

###### TDBXMetaDataCommand.Destroy

FQueryCommand.DisposeOf instead of FreeAndNil(FQueryCommand) ??? link to DisposeOf

##### Data.DbxSqlite.pas

###### Command.Free changed to Command.DisposeOf

###### TDBXSqlLiteConnection

Throws exception on failed driver initialization

##### Data.SqlExpr.pas

###### TSQLConnection.FreeSchemaTable

DataSet.DisposeOf instead of DataSet.Free

#### EMS

##### DesktopConsole

###### Many new methods and properties

##### EMS.ResourceAPI.pas

###### New TEndpointRequestBody.JSONReader

###### New TEndpointResponseBody.JSONWriter

###### TEndpointContext

New nested TEdgemodule class

New EndpointName property

New Edgemodule

###### TEndpointRequest

Removed Endpoint property

##### EMS.ResourceTypes.pas

###### Removed Endpoint parameter classes

##### EMS.Services.pas

###### New module interfaces

/// <summary>Methods for executing module resource requests from custom endpoint implementations.</summary>

IEMSModuleResourceRequest = interface(IEMSResourceRequest)

function GetModuleName: string;

/// <summary>Get the name of the module.</summary>

property ModuleName: string read GetModuleName;

end;

/// <summary>Service for executing module resource requests from custom endpoint implementations.</summary>

IEMSModuleResourceRequestService = interface

['{24772850-762F-43FA-9B9E-0818750F0E06}']

function CreateRequest(const AModuleName: string): IEMSModuleResourceRequest;

end;

###### TEMSInternalAPI

TSegmentNames new string const

Edgemodules

Resources

Many new module and module resource management functions

Unregister

Query

Retrieve

Update

Delete

Get

Patch

Post

Put

#### FireDAC

##### General

###### Free -> FDFree

###### FreeAndNil -> FDFreeAndNil

###### Longint -> Integer

###### PLongint -> PInteger

###### LongWord -> Cardinal

##### FireDAC.ADO.Import.pas

###### Internal BCD changed to dtFMTBcd

##### TFDBatchMoveDataSetDriver.GetTableDefs

###### Added support for Provider.PSGetKeyFields()

###### Improved retrieval of TIndexDefs

##### TDFBatchMoveMappings.GetKeyValues

###### Result is now assigned (bug fix)

##### TFDBatchMoveSQLWriter

###### new CreateTableParts : TFDPhysCreateTableParts

The CreateTableParts property specifies which table definition parts will be included into CREATE TABLE statement and which associated database objects will be create together with table.

property CreateTableParts: TFDPhysCreateTableParts read FCreateTableParts write FCreateTableParts default [tpTable .. tpIndexes];

##### TFDCustomConnection

###### Promoted to protected

AttachClient

DetachClient

###### New ExecSQL overload

function ExecSQL(const ASQL: String; var AResultSet: TDataSet): LongInt; overload;

##### TFDCustomSchemaAdapter

###### New UpdateOptions property

/// <summary> UpdateOptions property returns reference to update options,

/// will be used at applying updated. </summary>

property UpdateOptions: TFDUpdateOptions read GetUpdateOptions write SetUpdateOptions;

##### TFDAdaptedDataSet

###### ServerDeleteAll is now virtual

##### TFDCustomMemTable

###### new protected DefineProperties override

procedure DefineProperties(AFiler: TFiler); override;

###### dropped property AutoCommitUpdated introduced private procedure ReadAutoCommitUpdated(AReader: TReader)

##### TFDCustomConnection.ExecSQL

###### Fixed AV when Params were nil

##### TFDTable.GetRecNo change

if not FetchOptions.LiveWindowParanoic or **(State = dsInsert) or**

(SourceView = nil) or (SourceView.Rows.Count = 0) then

Exit;

##### TFDDataSet

###### Removed Resync

###### Changed Reconcile

function Reconcile(**ASuccess: Boolean = True**): Boolean;

###### StartWait

Performance improvement with nested procedure

###### StopWait

Performance improvement with nested procedure

###### Error local procedure (DB.pas ?).DatabaseErrorFmt call requires compiler magic calls for strings ref / deref. Because in most cases the method will work without errors (optimistic approach), then these ref / deref are extra work for CPU. Moving DatabaseErrorFmt call into local procedure also moves ref / deref into this local procedure. So, this will slightly improve performance.

###### SetUpdateFieldAttributes

Exits if FDataLoadSave <> nil

###### GetParentRow

Ensures Record buffer for parent is established

###### GetRecNo

Returns -1 if row is in insert state

###### GetFieldColumn

Introduced nested function to clean up logic

###### InternalPost

StartWait and StopWait moved to the edges of the routine

###### InternalRefresh

Refreshing speed up by removing temporary indices

###### CopyDataset

Refactored to clean up code

###### DoReconcileErrorHandler

Default AAction result is now raSkip instead of raMerge

###### ApplyUpdated

Now only updated if MaxErrors is not exceeded

###### Reconcile

Added boolean flag for success. If true and AutoCommit, AcceptChanges

Also calls UpdateCursorPos now

###### BeginBatch and EndBatch now update WaitCounter

##### TFDScriptParser.ExtractCommand

###### Added #0, #13, #10 handling in several places

##### IFDDAptTableAdapter

###### Removed property AutoCommitUpdated and getter/setter

##### IFDDAptSchemaAdapter

###### Added property Options : IFDStanOptions

##### TFDDAptTableAdapter

###### Removed property AutoCommitUpdated and getter/setter

##### TFDDAptSchemaAdapter

###### Added property Options : IFDStanOptions

##### TFDDatSColumn.GetStorageSize

###### Performance improvement, caching storage size rather than re-determining it

##### TFDDatSColumnList.FixAutoIncs

###### Nested procedure performance improvement

##### TFDDatSRow

###### CheckRowConstraints

Performance improvement: another early exit for SkipCheckConstraint

###### EndEdit

**if not ANoVersion then begin**

FpCurrent := FpProposed;

FpProposed := nil;

FRowState := rsModified;

**end**

**else begin**

**FreeBuffer(FpOriginal);**

**FpOriginal := FpProposed;**

**FpProposed := nil;**

**FRowState := rsUnchanged;**

**end;**

###### Removed SetBlobLength

###### refactored SetBlobData

###### InternalSaveToStorage

**if FpOriginal <> nil then** SaveRowVersion(AStorage, 'Original', FpOriginal, rsImportingOriginal);

**if FpCurrent <> nil then** SaveRowVersion(AStorage, 'Current', FpCurrent, rsImportingCurent);

**if FpProposed <> nil then** SaveRowVersion(AStorage, 'Proposed', FpProposed, rsImportingProposed);

##### TFDPhysADSMetadata

###### Added AFreeConn: Boolean to constructor

###### GetIsFileBased

Returns FreeConnection value rather than True

##### TFDPhysADSCommandGenerator

###### Added GetMerge override

##### TFDPhysASACommandGenerator

###### Added GetMerge override

###### GetLimitSelect checks for Rows <> 0

##### TFDPhysDB2Connection.SetupConnection

if oParams.ExtendedMetadata then

ODBCConnection.DESCRIBE\_OUTPUT\_LEVEL := **3** (instead of '3')

**else**

**ODBCConnection.DESCRIBE\_OUTPUT\_LEVEL := 2;**

##### TFDPhysDb2CommandGenerator.GetLimitSelect

if ASkip + ARows <> MAXINT then

**// DB2: does not support FETCH FIRST 0 ROWS ONLY and returns**

**// "The numeric literal 0 is not valid because its value is out of range"**

**if ARows = 0 then**

**Result := 'SELECT \* FROM (' + BRK + ASQL + BRK + ') A' + BRK + 'WHERE 0 = 1'**

**else**

Result := ASQL + BRK + 'FETCH FIRST ' + IntToStr(ASkip + ARows) + ' ROWS ONLY'

##### TDSCustomDelphiProxyWriter.WriteMethodImplementation

###### Fixed syntax error generation

###### Free -> FDFree in proxy

##### TFDPhysIBCommandBase.CreateColInfos

###### Added support for dtGUID

##### TFDPhysIBCommandGenerator

###### Added GetMerge override

###### GetColumnType

**case AColumn.DataType of**

**dtBoolean:**

**begin**

**Supports(FConnMeta, IFDPhysIBConnectionMetadata, oIBConnMeta);**

**if (oIBConnMeta.Brand = ibFirebird) and (FConnMeta.ServerVersion >= ivFB030000) or**

**(oIBConnMeta.Brand = ibInterbase) and (FConnMeta.ServerVersion >= ivIB090000) then**

**Result := 'BOOLEAN'**

**else**

**Result := 'SMALLINT';** (was 'Char(1)')

**end;**

##### TIBLib

###### Added Unload override

##### TIBVariable

###### Added property FDDataType: TFDDataType

###### GetData changed processing for dtGUID

###### SetData changed processing for dtGUID

##### TIBEvents.Stop

###### Supports waiting for timeout

##### TIBStatement.GetSQLInfoStr

buff: array [0..**16384**] of Byte; (was 1024)

##### TFDPhysInfxConnection

###### Knows if transactions are supported for the connection

##### TFDPhysInfxConnectionDefParams

###### Added TxSupported: TFDInfxTxSupported (tsYes, tsNo, tsChoose);

##### TFDPhysInfxMetadata

###### Added TxSupported to Constructor

###### GetTxSavepoints returns TxSupported instead of True

##### TFDPhysInfxCommandGenerator.GetLimitSelect

**// Informix: does not support FIRST 0 and returns "syntax error"**

**if ARows = 0 then**

**Result := 'SELECT \* FROM (' + BRK + ASQL + BRK + ') A' + BRK + 'WHERE 0 = 1'**

**else** if HasKW(s, 'UNION') or HasKW(s, 'MINUS') or HasKW(s, 'INTERSECT') or HasKW(s, 'EXCEPT') then

Result := 'SELECT ' + GetLimit + ' \* FROM (' + BRK + ASQL + BRK + ') A'

##### TFDPhysMergeAction = (maInsertUpdate, maInsert, maUpdate); -> TFDPhysMergeAction = (maInsertUpdate, maInsertIgnore);

##### New driver: MongoDB

##### FireDAC.Phys.MSSQLCli

###### Conditional compilation for MSWINDOWS

**{$IFDEF MSWINDOWS}**

**uses**

**Winapi.Windows;**

**{$ENDIF}**

and

**{$IFDEF MSWINDOWS}**

**FSCTL\_SQL\_FILESTREAM\_FETCH\_OLD\_CONTENT =**

**(FILE\_DEVICE\_FILE\_SYSTEM shl 16) or (FILE\_ANY\_ACCESS shl 14) or (2392 shl 2) or (METHOD\_BUFFERED);**

**{$ENDIF}**

##### TFDPhysMSSQLCommandGenerator

###### Added GetMerge override

##### TMSSQLFileStream.Create

###### Special handling on MSWINDOWS

**{$IFDEF MSWINDOWS}**

**//** [**https://msdn.microsoft.com/en-us/library/cc627407(v=sql.100).aspx**](https://msdn.microsoft.com/en-us/library/cc627407(v=sql.100).aspx)

**if iAccess = SQL\_FILESTREAM\_READWRITE then**

**if not DeviceIoControl(Handle, FSCTL\_SQL\_FILESTREAM\_FETCH\_OLD\_CONTENT,**

**nil, 0, nil, 0, iBytesReturned, nil) then**

**Error;**

**{$ENDIF}**

##### TFDPhysMySQLConnection InsertID now tracked by connection, not session

##### TFDPhysMySQLCommandGenerator added GetMerge override

##### TFDPhysODBCConnection.InternalCreateMetadata Added support for new MetaData constructor paraemters

##### TFDPhysODBCCommand.FD2CDataType

###### Added dkPervasive as SQL\_C\_CHAR for SQL\_C\_NUMERIC

##### TFDPhysODBCConnectionDefParams.NumericFormat storage default now nfString (was nfBinary)

##### TFDPhysODBCMetadata added override for GetIsFileBased

##### TODBCDriverKind added dkPervasive

##### TODBCSelectItem added property TYPE\_NAME: string

##### TODBCConnection property DESCRIBE\_OUTPUT\_LEVEL is now Integer (was String)

##### TODBCConnection.AfterConnect

###### added better handling for DB2

###### detects supports dkPervasive

##### TFDPhysTransaction.LockAutoStop

Inc(FLockAutoStopCount);

**if (FLockAutoStopCount = 1) and not GetActive then begin**

**StartTransaction;**

**FExplicitActive := False;**

**end;**

##### TFDPhysPgTransaction.InternalNotify has additional notification types

##### TFDPhysPgCommandGenerator added GetMerge override

##### TFDPhysCommandGenerator added

###### function GetInsert(const AOption: String): String; virtual;

###### function GetUpdate: String; virtual;

###### function GetMerge(AAction: TFDPhysMergeAction): String; virtual;

##### TFDPhysSQLiteCommandGenerator

###### added GetMerge override

###### GetColumnType updated for primary key detection

##### TSQLiteVDSTable ReleasePoints updated savepoint handling

##### TFDPhysTDataCommandGenerator

###### added GetMerge override

###### GetLimitSelect

**// Teradata: does not support TOP 0 and returns "syntax error"**

**if ARows = 0 then**

**Result := 'SELECT \* FROM (' + BRK + ASQL + BRK + ') A' + BRK + 'WHERE 0 = 1'**

**else** if HasKW(s, 'UNION') or HasKW(s, 'MINUS') or HasKW(s, 'INTERSECT') or

HasKW(s, 'EXCEPT') or HasKW(s, 'DISTINCT') then

##### TFDPhysTDBXConnection.InternalCreateMetadata supports additional Metadata constructor parameters

##### TFDPhysTDBXCommand.GetParamValues

###### supports streams

###### updated ftOraBlob handling

##### TFDStanAsyncExecutor.Run for POSIX, Sleep(1) instead of Sleep(0)

##### FireDAC.Stan.Consts updated for version and MongoDB

##### FireDAC.Stan.Error added MongoDB error handling

##### TFDExpressionToken added etLCBracket, ctRCBracket

##### TFDExpressionParser updated

##### TFDRDBMSKinds

###### MongoDB added (16)

###### Other now 17 (was 16)

##### TFDStringList added property UseLocale: boolean

##### TFDStorableObject.LoadFromStorage performance improvement (one call to GetObjectStyle)

##### TFDStorableObject.SaveToStorage performance improvement (one call to GetObjectStyle)

##### TFDUpdateOptionValue added uvAutoCommitUpdated

##### TFDUpdateOptions added property AutoCommitUpdated : boolean

##### TFDFormatOptions.ResolveDataType bug fix on name mask and type mask comparisons

##### TFDFormatOptions.ColumnDef2FieldDef

dtInt64**~~, dtUint64~~**:

ADestFieldType := ftLargeint;

**dtUInt64:**

**ADestFieldType := {$IF DEFINED(CPUARM64)} ftLongWord {$ELSE} ftLargeint {$ENDIF};**

and

ftLongWord:

ADestDataType := **{$IF DEFINED(CPUARM64)} dtUInt64 {$ELSE}** dtUInt32 **{$ENDIF}**;

##### TFDParam lots of changes has many new inline functions

##### FireDAC.Stan.ResStrs new MongoDB resource strings

##### TFDReadSharedMemoryStream

###### Removed Destroy

###### Added Realloc

##### FDGetAppPath uses TPath.GetHomePath

##### FDGetTempPath uses TPath.GetTempPath

##### TfrmFDGUIxFormsFormatOptions.LoadFrom added TypeMask

#### REST

##### Rest Debugger

###### Added scrollbox

###### Added enter key to open on URL

##### OAuth support cleaned up

##### EMS

###### TEMSClientAPI

Added support for module management

Added support for module resource management

QueryResource added a TProc parameter

DeleteResource added a TProc parameter

AddResource added a TProc parameter

###### EMS Meta Types

New TMetaModule

New TMetaModuleResource

New TMetaFoundModule

New TMetaFoundModuleResource

TEMSMetaFactory

New CreateMetaFoundModule

New CreateMetaFoundModuleResource

###### TEMSQueryAPI

GetServiceNames added Modules and ModuleResources

Query added Modules and ModuleResources

###### new IBackendModuleName

###### new IBackendModuleResourceName

##### new function TParseStorageAPI.GetJsonDateFormat : TJsonDateFormat

##### TBackendQueryServiceNames added Modules and ModuleResources

##### IBackendStorageAPI2 for custom date format objects

##### TBackendStorageAPI

###### added function AddDateOption (ADateFormat: TJSONDateFormat; AOptions: TJSONOptions): TJSONOptions;

###### CreateObject<T> supports custom date format

###### FindObject<T> supports custom date format

###### QueryObjects<T> supports custom date format

###### UpdateObject<T> supports custom date format

##### TRESTRequestParameterList new overloads for AddItem

##### new TBodyParams class

##### TCustomRESTRequest

###### new nested TBody class, wrapper around body request parameter

###### new property Body : TBody

###### Execute uses Body parameters now

###### AddBody overloads are just a wrapper for TBodyParams.Add() now

###### ClearBody calls FBody.ClearBody

###### ResetToDefaults calls FBody.ClearBody

##### TCustomRestResponse new property JSONReader: TJSONTextReader

##### TCustomAuthenticator

###### removed IAuthenticator

###### procedure Authenticate no longer virtual

### DB

#### TFields added a TDictionary<string, TField> for field name lookup! Using AnsiLowerCase for case-insensitivity.

##### Add() updated FDict

##### Remove() updated FDict

##### ClearBase() updated FDict

##### FindField() list iteration replaced with FDict.TryGetValue()

##### FieldByName() performance improvement with nested procedure Error

#### TADTField added Assign override

#### TDataSet

##### new function GetCalcFieldTypes: TFieldTypes; virtual;

##### BindFields() uses GetCalcFieldTypes()

##### Resync() now checks for Active before doing anything

#### new FieldTypeVarMap definition

(embed FieldTypeVarMap.png)

#### TDefaultLookupList.ValueOfKey Should be using Dict as well?

#### TDBBitConverter class procedures

##### UnsafeFromVariant() added support for variant array

##### UnsafeInToVariant() assigns return value

##### UnsafeInToInterface() assigns return value

#### TField.GetAsBCD() uses StrToBCD(GetAsString) instead of CurrToBCD(GetAsCurrency)

#### function TWideStringField.GetAsBytes: TArray<Byte> uses WideBytesOf() instead of BytesOf()

#### TIntegerField.GetValue() uses assigned IOBuffer to convert to Integer. ??? look at TDBBitConverter.UnsafeFrom<Integer>(0, FIOBuffer);

#### TDateTimeField.GetValue() uses

if FIOBuffer <> nil then

TDBBitConverter.UnsafeFrom<Double>(0.0, FIOBuffer);

#### TFMTBCDField.GetValue() improves decimal determination

#### TDataSetField.Assign() no longer calls inherited

#### TDataSource.DataEvent() no longer calls OnDataChange when DataSource is being destroyed

#### TParams.ParamByName() performance improvement with nested "not found" error procedure

#### function GetIndexForOrderBy() skips "order by" resolution attempt when a number in the order by SQL expression is < 1

#### CurrToBCD changes Bcd := TBcd(CurrToStr(Curr)) to NormalizeBcd(TBcd(CurrToStr(Curr)), Bcd, Precision, Decimals);

## Live Bindings Editors

### TBindListGridEditor.SetRowCount() retains setting for cell in edit mode

### TListViewItemEditorObject.CurrentItem() now supports IListViewFilterable

### TBindListViewEditor.AddItem() now supports IListViewFilterable

## DUnit and DUnitX are updated

## FlatBox2D updated, lots of helpful comments

## FireMonkey

### Almost every source file has been touched. Many are unit reference changes

### New Windows native controls

### New ListView adapters

### New ListView appearances

### TActionLink added property HintLinked : boolean

### TAction property OnHint is published

### TCustomAction.Execute() boolean value determination modified

### TAndroidBannerAd() now uses TAndroidHelper.Activity instead of SharedActivity

### TiOSBannerAdDelegate.bannerViewActionDidFinish() no longer calls FWinService.ShowWindow()

### TBannerAd supports explicit iOS 32 and 64 bit targets

### TAnimation

#### added property NormalizedTime : Single

#### ProcessTick() supports AutoReverse

#### Start() supports AutoReverse

#### Stop() supports AutoReverse

#### StopAtCurrent() supports AutoReverse

### BehaviorManager removed TOSPlatform

### TCalendarModel added DefaultTodayDefault and DefaultWeekNumbers, both False

### TCustomCalendar introduced as parent for TCalendar

### TCalendar publishes:

#### TouchTargetExpansion

#### ShowHint

### TiOSNativeCalendar added property Model : TCalendarModel

### TiOSCalendarProxy removed

### TStyledCalendar has lots of new functionality

### TStyledCalendarProxy removed

### TCalendarEditModel

#### Overrides GetDefaultStyleLookupName()

#### Overrides constructor

### TStyledCalendarEdit. GetDefaultStyleLookupName() override removed

### TStyledCalendarEditProxy removed

### public SetMatrix() override replaced with protected DoSetMatrix() override

#### TCanvas (virtual)

#### TD2DBitmapHandle

#### TCanvasGdiPlus

#### TCanvasQuartz

### TCanvasGdiPlus.DoEndScene() updated

### TComboColorBox.ApplyStyle() uses StyledControl instead of TFMXObject

### TColorButton.ApplyStyle() uses TShape instead of TFMXObject

### TComboEditBase new procedure CloseDropDown

### published ParentShowHint

#### TCalendar

#### TComboEdit

#### TComboTrackBar

### TStyledComboEdit

#### PMDropDown(), PMCloseDropDown(), CloseDropDown() moved to TStyledComboEditBase

#### KeyDown() updated processing of Enter key

// Mark Return key as processed to avoid handling of Default button?

if (Key = vkReturn) and ((Shift = [ssAlt]) or (Shift = [])) then

begin

Key := 0;

KeyChar := #0;

end;

### TStyledComboEditProxy removed

### TStyledComboTrackBar overrides CloseDropDown()

### Hints

#### new IHintReceiver

#### new IHintRegistry

#### new THint

#### TControlAction publishes Hint and OnHint

#### new THintAction

#### new TSysCommonAction published property OnHint

#### new IControl functions

##### HasHint: boolean

##### GetHintString : string

##### GetHintObject : TObject

### TControl

#### new property AbsoluteClipRect: TRectF

#### new property ApplyingEffect: Boolean

#### new property Pressed: Boolean

#### property PressedPosition: TPointF when MouseDown is called

#### new property ShowHint: Boolean

#### published property Hint : string

### TStyledControl

#### new function FindStyleResource<T>() overloads

#### new class property LoadableStyle: TFmxObject

#### class function LookupStyleObject()

### new TStyleChangedMessage class

### new TStyleCollectionItem class Collection item that's stored one style for specified platform

### new TStyleCollection class stored styles for different platform

### new TStyleBookDesignInfo record for Design-time only data

### TStyleBook has new public methods and properties

### TTextControl new property PrefixStyle: TPrefixStyle ???

### Presentation

#### New constants

#### PM\_ROOT\_CHANGED = PM\_BASE + 56;

#### PM\_MOUSE\_WHEEL = PM\_BASE + 57;

##### PM\_GET\_FIRST\_CONTROL\_WITH\_GESTURE = PM\_BASE + 58;

#### new TMouseWheelInfo, used for transfering mouse wheel event into presentation in design time

#### new TFirstControlWithGestureInfo, used for requesting control, which supports specified gestures

#### TPresentedControl

##### new procedure LoadPresentation; virtual;

##### new procedure UnloadPresentation; virtual;

##### new procedure ReloadPresentation;

##### new property Presentation: TObject, return instance of presentation object

##### new property PresentationState : TPresentationState, which is (NotLoaded, Loading, Loaded, Unloading);

### TControl3D

#### added protected methods

##### function HasHint: Boolean; virtual;

##### function GetHintString: string; virtual;

##### function GetHintObject: TObject; virtual;

#### SetOpaque() only calls repaint if value is changed

### TCustomDateTimeEdit added protected procedure DoTextObjectSizeChanged(Sender: TObject); virtual;

### TTimeEdit and TDateEdit publish

#### TouchTargetExpansion

#### ParentShowHint

#### ShowHint

### TiOSNativeEdit

#### new protected method messages for Caret position and text Selection

#### new procedure SelectionChanged()

#### new function canPerformAction()

#### new function SelectedRange: NSRange

### TEdit, TEditButton, TSpinEditButton, TComboBox, TMemo publish

#### ParentShowHint

#### ShowHint

### TStyledEdit new Max Length changed message procedure

### TStyledEditProxy removed

### new TWinNativeEdit

### TStyledEditBoxProxy removed

### Internal updated to FontGlyphs

### Windows IDesignerHook procedure SupportsPlatformService removed

### TApplication added Hint support

### TCommonCustomForm added Hint, Scale and Style change support

### TForm publishes ShowHint

### TFrame publishes ParentShowHint, ShowHint

### TBitmapCodecAndroid.IsValid() updated

### new TCanvas property Offset : TPointF allows offset to drawing area

### new Android TUIThreadCaller class procedures

### new iOS helper functions, BitmapSurfaceToUIImage() and UIImageToBitmapSurface(),FontToCTFontRef()

### new Windows helpers

#### TImeModeHelper

#### TWow64Redirection

#### some bitmap functions

### TInAppPurchase supports iOS 32 and 64 bit

### TCustomScrollBox internal changes

### TListBox selection enhancements

### ListView has major refactoring

#### new TAdapterListView is the base class for TListView controls. It provides the basic interface between TListView and TListViewAdapter. TListViewAdapter events are intercepted at this level and should be implemented by the descendants in their corresponding Do... virtual methods

#### TCustomListView is now class(TAppearanceListView)

#### TListItemObject is now TListItemDrawable

#### new TAbstractListViewAdapter

#### new TListViewItems, a minimal concrete implementation of IListViewAdapter

#### new TFilterableListViewItems

#### new IAppearanceItemOwner

#### new IPublishedAppearanceOwner

#### new TAppearanceListViewItems

#### new TListViewItem

#### new TObjectAppearance

#### new TCommonObjectAppearance

#### new TCustomTextObjectAppearance

#### and so on

#### iOS TableView updated

### Maps

#### TMapView publishes

##### property Margins

##### property LayerOptions

#### iOS Maps routines have updated

### MediaLibrary updated

#### TImageManagerAndroid added const AWriteImageCompletionEvent: TWriteImageCompletionEvent = nil parameter to AddImageToSavedPhotosAlbum()

#### iOS

### Memo

#### TiOSNativeMemo improves text selection

#### TMemo

##### published property OnPresentationNameChoosing;

##### unpublished property OnHScrollChange and OnVScrollChange

#### TStyledMemo gets messages for Lines Changed, Max Length Changed, and text selection

#### new TWinNativeMemo

### TCaretPosition has new overrides for Equal() and NotEqual() operators

### iOS multi touch logic updated

### TMultiViewPresentation

#### now class abstract(TInterfacedObject)

#### new procedure ParentChanged; virtual

#### new function CanShowHideInDesignTime: Boolean; virtual

#### new property ExpandedSize: TSizeF

#### new property CollapsedSize : TSizeF

### new TMultiViewMode = (PlatformBehaviour, Panel, Popover, Drawer, Custom, NavigationPane);

### new TCustomMultiView property NavigationPaneOptions : TNavigationPaneAppearance (a new TMultiViewAppearance descendant)

### TLine has new properties

### New TAndroidDateTimePicker support for NativeDatePicker and NativeTimePicker

### New Platform structure/routine updated for iOS, Mac, Windows

### new TPresentationProxyFactory methods

### Updated to TiOSNativeView

### New TWinView, TWinPresentation

### New TWinPresentationProxy<T: TWinPresentation, constructor> = class(TPresentationProxy)

### TScrollBox

#### updated TiOSScrollBox

#### new TWinScrollBox

### New TTouchAnimationAdapter Helpers class that encapsulates touch animation behavior, should be attached to other style object

### TSwitch

#### iOS updated

#### new TWinSwitch

### TTabControl updated (and TTabItem)

### TTextService

#### new property MaxLength : Integer

#### new property MultiLine : boolean

### TFMXObject uses TEnumProcResult (Continue, Discard, Stop)

### WebBrowser minor updated

#### Android

#### Cocoa

#### iOS

#### Mac

#### Win

### New TWinZOrderManager

### New FMX StyleLookup property editors

## InterBase Express

### Integer type changes

### New source files

### Refactored code

### New TIBFieldHelper.ChangeState returns the change state for the field for InterBase subscriptions

## Internet

### TWebBrowser

#### Many protected and private changes

#### Publishes property StyleElements

#### New property OnShowScriptError : TWebBrowserShowScriptError

### TWebRequest new procedure ReadTotalContent()

### TISAPIRequest

#### overrides procedure ReadTotalContent()

#### GetStringVariable() speeds up RawContent support

### TDispatchFileRequest.QualifyFileName() cleaned up

## RTL

### Android

#### Androidapi.AppGlue.pas

##### improves error logging

##### enhances app startup

##### added assignment to System.DelphiActivity

##### set System.DelphiActivity to nil on app free

#### Many android api helpers inlined, others deprecated

#### TAndroidHelper introduced

#### Androidapi.IOUtils

##### function GetExternalDocumentsDir : string added

##### function GetSharedDocumentsDir : string added

#### Many bluetooth enhancements

#### New Androidapi.JNI.Embarcadero.pas interfaces

#### Jbuild\_VERSION\_CODESClass interface added

##### KITKAT\_WATCH : Integer

##### LOLLIPOP : Integer

##### LOLLIPOP\_MR1: Integer

#### JToast interface added

#### JNIBridge

##### TJavaLocal updated

##### TClassLoader updated

#### New Android services

##### TAndroidBaseService (TDataModule)

##### TAndroidService

##### TAndroidIntentService http://developer.android.com/reference/android/app/IntentService.html

##### TServiceCallbacks class methods to connect native callbacks with the delphi service

##### TLocalServiceConnection Helper class to manage a connection to an android Local Service

##### TRemoteServiceConnection Helper class to manage a connection to an android Remote Service

##### TServiceApplication

#### TPath.GetSharedDocumentsPath uses shared instead of external

### iOS

#### TiOSLocationSensor.GetAuthorized() now supports kCLAuthorizationStatusAuthorizedWhenInUse

#### Address Book support

##### Interface to library

##### Address Book UI delegates

#### UIPasteboard new function &string: Pointer

#### UIAlertViewDelegate new procedure didPresentAlertView(alertView: UIAlertView)

#### UITextField

##### new function selectedTextRange: UITextRange

##### new procedure setAttributedText(attributedText: NSAttributedString)

##### new function attributedText: NSAttributedString

##### new function beginningOfDocument: UITextPosition

##### new function offsetFromPosition(from: UITextPosition; toPosition: UITextPosition): NSInteger

##### new function endOfDocument: UITextPosition

##### new function positionFromPosition(position: UITextPosition; offset: NSInteger): UITextPosition

##### new function textRangeFromPosition(fromPosition: UITextPosition; toPosition: UITextPosition): UITextRange

### Common

#### Background Notifications

##### TNotification base

##### TBaseNotificationCenter

##### TCustomNotificationCenter(TComponent)

##### TNotificationCenter(TCustomNotificationCenter)

##### TPlatformNotificationCenter

###### Android

###### iOS

###### Mac

###### Windows

##### TMessageReceivedNotification

###### Android

###### iOS

#### TCustomSensor (and descendants) supports Stop, Start, and tracking Started (common functionality)

#### Performance improvements, new functions for TCharacter record

#### TCharHelper added IsDefined() overloads

#### pidWinARM platform identifier added

#### ThreadProc() slightly modified on Android

#### TListHelper internal management improvements

##### Managed items

##### Range deletion

##### Weak references

#### Binary comparer explicitly for TThreeByteData

#### CPUX64 comparers for 8 byte data

#### Hashing

##### New overload for THash.ToBigEndian(UInt64)

##### THashMD5

###### New inlines

###### new GetHMACAsBytes() overloads

##### THashSHA1

###### New inlines

###### new GetHMACAsBytes() overloads

###### new GetBlockSize()

###### new GetHashSize()

##### New THashSHA2

##### THashBobJenkins

###### New parameter for Reset(AInitialValue: Integer = 0)

###### new inlines

#### TFile.IntegerToFileAttributes performance improvement

### JSON

#### New TParseDateTimeInterceptor for conversions between TObject and TDateTime

#### New TJsonOption joDateFormatParse

#### TJson

##### JsonToObject overloads support joDateFormatParse

##### ObjectToJsonObject supports joDateFormatParse

##### ProcessOptions uses DisposeOf instead of Free for Json pairs

#### TJsonDates added Parse to TFormat

#### TJsonDateFormat added jdfParse

#### TParseDate represents a Parse Date Object type for marshalling and unmarshalling

#### TJsonDates.AsDateTime supports Parse

#### function TTypeMarshaller<TSerial>

##### GetTypeConverter

###### added nil check

###### supports jdfParse

##### MarshalValue supports TListHelper on tkRecord

##### MarshalData added nil check

##### MarshalSimpleField added nil check

#### TJSONUnMarshal

##### CreateObject fixed memory leak on error

##### SetFieldArray supports tkRecord and TListHelper

#### TCustomJSONDataSetAdapter.AddJSONDataRow added handling of discovered Date/Time string representations

#### TCustomRESTResponseDataSetAdapter.Notification added notification for JsonValueChanged

#### Support for BSON http://bsonspec.org/

##### TBsonType

##### TBsonWriter

##### TBsonReader

#### JSON collection support

##### New TJSONCollectionBuilder wrapper around TJSONWriter to build a JSON array or object, with support for nested arrays and nested objects

###### nested TElements

###### nested TPairs

###### nested TBaseCollection

###### nested TParentCollection

###### nested TParentType = (None, Elements, Pairs)

##### New JSONArrayBuilder Call TJSONWriter to build a JSON array, with support for nested arrays and JSON objects

##### New JSONObjectBuilder Call TJSONWriter to build a JSON object, with support for nested arrays and JSON objects

##### new TJSONIterator

#### New TJONPathParser parses a JSON path with names and indices such as 'name', '[0]', 'entities.urls[0].indices[1]'

##### Converted from implementation-level TPathStringParser

##### FindJSONValue uses TJSONPathParser

#### TJsonObject

##### New nested TJSONParseOption = (IsUTF8, UseBool);

##### new TJSONParseOptions = set of TJSONParseOption;

###### IsUTF8 - data should be treated as UTF8

###### UseBool - Parser should create a TJSONBool object for each instance of "true" or "false" seen in the JSON data

##### HexToDecimal() now inlined

##### New overloads and inlines for ParseJSONValue()

##### UseBool : Boolean = False added to Parse()

##### TJSONBool now parent for TJSONFalse and TJSONTrue

#### New Readers/Writers

##### TJsonLineInfo Base class to provided information about the JSON position being written/read in a stream

##### TJsonReader

##### TJsonTextReader

##### TJsonTextUtils

##### TJsonWriter

##### TJsonTextWriter

##### TJsonObjectWriter

##### Many more JSON constants for error messages

#### New JSON types

##### TJsonToken

##### TJsonOid

##### TJsonRegEx

##### TJsonCodeWScope (BSON Code With Scope)

### RTTI PassByRef() updated for CPUARM64 on tkArray

### TThreadPool

#### New property Current : TThreadPool (used in TParallel)

#### QueueWorkItem() checks for ThreadPool = Self before using Local Queue

#### ThreadPool pointer info added to named thread for debugging

### TTask

#### New protected property ThreadPool : TThreadPool

#### ExecuteReplicates() Calls Complete(False) on exception

#### SetComplete() uses TInterlocked.CompareExchange<TLightweightEvent>(FDoneEvent, nil, nil)

#### GetDoneEvent() logic updated

#### HandleException() avoids possible AV

#### InternalAddComplete() avoids possible AV

#### InternalExecute() is smarter about completion

#### ProcessCompleteEvents() uses TMonitor

#### RemoveCompleteEvent() improves TMonitor usage

### HighDPI support

#### function IsDPIAware: boolean

#### function SetHighDPIAware: boolean

#### SetHighDPIAware called on initialization if running on Windows

### TEventDispatch changed from class(TObject, IUnknown, IDispatch) to class(TInterfacedObject, IDispatch)

### TDelegatedOleControl.PreMessageFilter() method signature changed

### New TShareContract (for Windows 8 and above)

### TWinTaskbar.InitTaskbarSupport() bug fix adding IID\_ITaskbarList4

### New Classes and Helpers for WinRT interfaces and strings

#### TInspectableObject Base Delphi object needed for Windows Runtime classes. Implements IInspectable interface

#### TWindowsString Helper needed to automate life cycle of HSTRING type and conversions from/to string

#### THStringHelper.ToString

#### WinRTClassNameAttribute

#### TWinRTImportHelper

#### TWinRTImport base class of all WinRT imports (many)

### TZipFile

#### new OnProgress : TZipProgressEvent

#### Add()

##### added file attributes parameter

##### supports Windows and POSIX file attributes

#### Extract() supports Windows and POSIX file attributes

### Beacons

#### New TBeaconAdvertiser beacon devices that can advertise proximity data

#### TCommonBeaconManager updated

#### TCustomBeacon.SetEnabled() performance tweak

#### TBeaconRegionItem.Assign() bug fix checks Source is TBeaconRegionItem

#### new TCustomBeaconDevice(Component)

#### new TBeaconDevice

#### TPlatformBeaconAdvertiser

##### Android

##### iOS

##### Mac

##### Declared but nil on Windows

#### iOS TBeaconRegion record

##### new Inside: boolean

##### new function BeaconBelongs(const ABeacon: IBeacon): Boolean; inline;

##### new function IsRegion(const AGUID: TGUID; AMajor: Integer; AMinor: Integer): Boolean; inline;

#### TiOSBeaconManager.UnRegisterBeacon() uses registered regions now

### Bluetooth

#### TBluetoothLE new property ForceRefreshCachedDevices: Boolean (only works on Android)

#### new TBluetooth16bitsUUID

#### new TBluetoothUUIDHelper Helper class for easy conversion of 128 bits Bluetooth UUIDs to 16 bits format, and back

#### new TBluetoothLEServiceData = class(TDictionary<TBluetoothUUID, TBytes>)

#### new TBluetoothLEAdvertiseData

#### TBluetoothLEManager

##### function StartDiscovery added ForceConnect: Boolean = false parameter

##### new property ForceRefreshCachedDevices: Boolean

#### TBluetoothLEDevice

##### new property ScannedAdvertiseData: TBluetoothLEAdvertiseData

##### new function Connect: Boolean

##### new property ForceRefreshCachedServices: Boolean Setting to True clears the internal cache and forces a refresh of the services from the remote device on service discovery process. This property only works on Android

##### new property OnConnect: TNotifyEvent

##### new property OnDisconnect: TNotifyEvent

##### property UpdateOnReconnect: Boolean only works on iOS/MacOS Setting to True updated the services tree when a previously connected device is re-connected. If is set to False, it is mandatory to redo a discovering services process again after reconnecting and previously obtained TBluetoothGattService, TBluetoothGattCharacteristic and TBluetoothGattDescriptor are no longer valid

#### TBluetoothGattServer new property AdvertiseData: TBluetoothLEAdvertiseData

### HTTP

#### TAndroidHTTPClient.DoExecuteRequest() added ValidateServerCertificate event

#### TAndroidHTTPRequest.DoPrepare() improves URL formatting

#### TMacHTTPRequest.RequestToNative() improves URL formatting

#### TMacConnectionDataDelegate.connectionWillSendRequestForAuthenticationChallenge() updated

#### THTTPClient.Execute() header processing cleaned up

#### THTTPClient.GetRequest()

##### uses specified request method

##### sets custom headers

#### THTTPResponse.InternalAddCookie() updated

#### THTTPResponse.GetContentLength() defaults to -1 instead of 0

#### TCookieManager.AddederverCookie() updated

#### TCookie.Create() bug fix removes URL decoding

#### TCookie.ToString() bug fix removes URL encoding

#### TWinHTTPClient.Create() uses WINHTTP\_ACCESS\_TYPE\_NO\_PROXY instead of WINHTTP\_ACCESS\_TYPE\_DEFAULT\_PROXY and later explicitly sets proxy if specified

#### Winhttp.dll used instead of wininet.dll

#### TNetHTTPRequest Execute() overload replaced with DoExecute()

#### TURI record has many updated and new properties

### OSX

#### NSDatePickerCellDelegate interface now has a procedure, datePickerCell()

#### NSView = interface(NSResponder)

##### new function fittingSize: NSSize;

##### new function intrinsicContentSize: NSSize;

#### NSTextField new procedure setPlaceholderString()

#### new procedure dispatch\_async\_f()

#### new NSNotificationCenter procedure removeObserver() overload

#### new procedure objc\_disposeClassPair(cls: Pointer)

#### TOCImport function GetObjectID inlined

### New POSIX TPThreadFunc = function(Parameter: Pointer): Pointer uses by Android and OSX

### new Windows function GetProductVersion(const AFileName: string; var AMajor, AMinor, ABuild: Cardinal): Boolean tries to retrieve the version info of the product with which a file was distributed and sets AMajor, AMinor and ABuild parameters according with this information. It returns False if it failed

### TRect

#### Contains() updated

#### IntersectsWith() updated

#### IntersectRect() updated

### Winapi

#### Direct3D10 updated

#### Imm updated

#### new KnownFolders

#### DPI

##### new WM\_DPICHANGED message

##### new TWMDpi record

##### new ShellScaling (DPI scaling)

#### new MsCTF import

#### new Mshtmhst import

##### IDocHostShowUI

##### IDocHostUIHandler

#### OpenGL constant updated

#### New Windows function imports

#### New Winrt (entire new folder)

### SOAP

#### TRefCache Performance improvement: TStrings now TDictionary

#### TSOAPDomConv dictionary speed ups

##### FindNodeByHREF()

##### AddMultiRefNode()

##### FindMultiRefNodeByID()

### Tethering has many updates

#### App Profile

#### Bluetooth Adapter

#### Core communication

#### Management

#### TCP protocol code streamlined

### ToolsAPI updated

#### DesignIntf TCustomModuleAttribute = (cmaVirtualSize, cmaMultiView, cmaFloating);

#### new interfaces

#### TSprig improvements

## VCL

### TSeStyle.ButtonDraw() updated

### TCheckListBox.DrawItem() style support updated

### ChangeScale() now overridden in:

#### TCalendar

#### TCoolBand

#### TCoolBar

#### TCustomHeaderControl

#### THeader

#### TTabbedNotebook

#### TTrackBar

#### etc.

### Win32NLSEnableIME() deprecated

### TControl.SetVisible(True) now calls AdjustSize if AutoSize is true

### TWinControl

#### WMInputLangChange() changed

#### ResetIme() changed

### Dialog updated for StyleManager

### StyleElements now published

#### TSplitter

#### TSpinButton

### TCustomForm DPI change aware

### new TCustomSharingContract(TComponent)

### new TSharingContract

### new TBrowseForFolder properties

### new TCustomStyle. GetElementDetails() overrides

### many new \*StyleHook classes

### TThemedMenuItem.DoDrawText() improved custom style support

### Theme updated

### New WinXCtrls for activity indicators

## OmniXML CharIs\_Char() improved