

Seagate Crystal Reports 7

API Additions and Changes for Seagate Crystal Reports 7 MR1

Overview

This document lists new additions and changes made to Crystal API constants, structures, and functions (PE calls) for Maintenance Release 1 (MR1). These changes are in addition to the added or modified functions and structure first introduced in the initial release of version 7 (search for SCR7_API_Changes.exe at <http://support.crystaldecisions.com/docs> for more information).

NOTE	Refer to the Technical Reference Guide (TechRefv2.pdf) on the Seagate Crystal Reports MR1 CD for a thorough description of the following new structures and functions
-------------	---

Contents

VERIFYING DATABASE	2
DATABASE TABLE LOCATION	4
WINDOW OPTIONS – TOOL TIPS	5
REPORT OPTIONS	6
PARAMETER PICK LISTS	7
PARAMETER RANGE/DISCRETE VALUES	8
NEW CONSTANTS FOR TIME AND DATETIME VALUES	9
NEW EVENTS	9
GRAPHING	10
OTHER RESOURCES	17
CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT	18

Verifying Database

Changes made to verifying databases include:

- Several new constants
- A new PTableDifferenceInfo structure
- A new PECheckNthTableDifferences function. This call is used to check to see if any changes were made to the table from its original state when the report was created. Note that this must be called before PEVerifyDatabase.

```
// Constants returned from PECheckNthTableDifferences.
These can be any
// combination of the following:
```

```
#define
PE_TCD_                                0x00000000
OKAY
#define
PE_TCD_
DATABAS                                0x00000001
ENOTFOU
ND
#define
PE_TCD_
SERVERN                                0x00000002
OTFOUND
#define
PE_TCD_
SERVERN                                0x00000004
OTOPENE
D
#define
PE_TCD_
ALIASCH                                0x00000008
ANGED
#define
PE_TCD_
INDEXES                                0x00000010
CHANGED
#define
PE_TCD_
DRIVERC                                0x00000020
HANGED
#define
PE_TCD_
DICTION                                0x00000040
ARYCHAN
GED
#define
PE_TCD_
FILETYP                                0x00000080
ECHANGE
D
#define
PE_TCD_
RECORDS                                0x00000100
IZECHAN
```

```
GED
#define
PE_TCD_
ACCESSC
HANGED
                                0x00000200
#define
PE_TCD_
PARAMET
ERSCHAN
GED
                                0x00000400
#define
PE_TCD_
LOCATIO
NCHANGE
D
                                0x00000800
#define
PE_TCD_
DATABAS
EOTHER
                                0x00001000
#define
PE_TCD_
NUMFIEL
DSCHANG
ED
                                0x00010000
#define
PE_TCD_
FIELDOT
HER
                                0x00020000
#define
PE_TCD_
FIELDNA
MECHANG
ED
                                0x00040000
#define
PE_TCD_
FIELDDE
SCCHANG
ED
                                0x00080000
#define
PE_TCD_
FIELDTY
PECHANG
ED
                                0x00100000
#define
PE_TCD_
FIELDSI
ZECHANG
ED
                                0x00200000
#define
PE_TCD_
NATIVEF
IELDTYP
ECHANGE
D
                                0x00400000
#define
PE_TCD_
NATIVEF
IELDOFF
SETCHAN
GED
                                0x00800000
#define
PE_TCD_
                                0x01000000
```

```

NATIVEF
IELDSIZ
ECHANGE
D
#define
PE_TCD_
FIELDDE           0x02000000
CPLACES
CHANGED

typedef struct PTableDifferenceInfo
{
WORD StructSize;
DWORD tableDifferences;           // any combination of PE_TCD_*
DWORD reserved1;                 // reserved - do not use
DWORD reserved2;                 // reserved - do not use
} PTableDifferenceInfo;

#define PE_SIZEOF_TABLE_DIFFERENCE_INFO (sizeof
(PTableDifferenceInfo))

// PECheckNthTableDifferences is not implemented for
reports based on a
// dictionary. The function then returns
PE_ERR_NOTIMPLEMENTED
// Note that PECheckNthTableDifferences must be called
before
// PVerifyDatabase

BOOL CRPE_API PECheckNthTableDifferences (short printJob,
short tableN, PTableDifferenceInfo FAR
*tableDifferenceInfo);

```

Database Table Location

There is now added support for Microsoft Access table names in the updated PTableLocation structure:

- The SubLocation member allows you to change to a different Microsoft Access table within the same .mdb when using a native connection. An example would be “c:\xtreme.mdb” as the Location, and “Customer” as the SubLocation.

Note that this member is only to be used for Microsoft Access databases.

- The ConnectBuffer member is used to provide connection information to any attached tables.

Newly added constants and structure members are indicated in bold:

```

#define PE_TABLE_LOCATION_LEN           256
#define PE_SIZEOF_TABLE_LOCATION      (sizeof
(PTableLocation))
#define PE_CONNECTION_BUFFER_LEN      512

```

```

typedef struct PTableLocation
{
    // Initialize to PE_SIZEOF_TABLE_LOCATION.
    WORD StructSize;

    // String is null-terminated.
    char Location [PE_TABLE_LOCATION_LEN];

    char SubLocation[PE_TABLE_LOCATION_LEN];

    //Connection Info for attached tables
    char ConnectBuffer[PE_CONNECTION_BUFFER_LEN];
}PTableLocation;

```

Window Options – Tool Tips

Tool tips members are now included in the PEWindowsOptions structure.

Newly structure members are indicated in bold:

```

typedef struct PEWindowOptions
{
    WORD StructSize;           // initialize to PE_SIZEOF_WINDOW_OPTIONS
    short hasGroupTree;       // BOOL value, except use PE_UNCHANGED
                               // for no change
    short canDrillDown;       // BOOL value, except use PE_UNCHANGED for // no
                               // change
    short hasNavigationControls; // BOOL value, except use PE_UNCHANGED for
                               // no change
    short hasCancelButton;    // BOOL value, except use PE_UNCHANGED for // no
                               // change
    short hasPrintButton;     // BOOL value, except use PE_UNCHANGED for // no
                               // change
    short hasExportButton;    // BOOL value, except use PE_UNCHANGED for // no
                               // change
    short hasZoomControl;    // BOOL value, except use PE_UNCHANGED for // no
                               // change
    short hasCloseButton;    // BOOL value, except use PE_UNCHANGED for // no
                               // change
    short hasProgressControls; // BOOL value, except use PE_UNCHANGED for // no
                               // change
    short hasSearchButton;    // BOOL value, except use PE_UNCHANGED for // no
                               // change
    short hasPrintSetupButton; // BOOL value, except use PE_UNCHANGED for // no
                               // change
    short hasRefreshButton;   // BOOL value, except use PE_UNCHANGED for // no
                               // change
    short showToolBarTips;    // BOOL value, except use PE_UNCHANGED for // no
                               // change.
                               // Default is TRUE (*Show* tooltips on
                               // toolbar)
    short showDocumentTips;   // BOOL value, except use PE_UNCHANGED for // no
                               // change.
                               // Default is FALSE (*Hide* tooltips on
                               // document).
} PEWindowOptions;

```

Report Options

The PEReportOptions structure now contains a doAsyncQuery member.

The DoAsyncQuery member is used to allow the process out on the database server to be stopped without having to hit the server for at least one record. When set to TRUE, the query will be done on the server asynchronously allowing the user to cancel the query before it completes. Note this is only available on database dlls which support asynchronous processing.

New structure members are indicated in bold:

```
typedef struct PEReportOptions
{
    WORD StructSize; // initialize to
    PE_SIZEOF_REPORT_OPTIONS
    short saveDataWithReport; // BOOL value, except
        // use PE_UNCHANGED for no change
    short saveSummariesWithReport; // BOOL value, except use PE_UNCHANGED
        // for no change
    short useIndexForSpeed; // BOOL value, except use PE_UNCHANGED for no // change
    short translateDOSStrings; // BOOL value, except use PE_UNCHANGED for no // change
    short translateDOSMemos; // BOOL value, except use PE_UNCHANGED for no // change
    short convertDateTimeType; // a PE_RPTOPT_ value, except use PE_UNCHANGED // for
        no change
    short convertNullFieldToDefault; // BOOL value, except use PE_UNCHANGED // for no
        change
    short morePrintEngineErrorMessages; // BOOL value, except use
        // PE_UNCHANGED for no change
    short caseInsensitiveSQLData; // BOOL value, except use PE_UNCHANGED for // no
        change
    short verifyOnEveryPrint; // BOOL value, except use PE_UNCHANGED for // no
        change
    short zoomMode; // a PE_ZOOM_ constant, except use
        // PE_UNCHANGED for no change
    short hasGroupTree; // BOOL value, except use PE_UNCHANGED for // no
        change
    short dontGenerateDataForHiddenObjects; // BOOL value, except use
        // PE_UNCHANGED for no change
    short performGroupingOnServer; // BOOL value, except use
        // PE_UNCHANGED for no change
    short doAsyncQuery; // BOOL value, except use
        PE_UNCHANGED for no change
}PEReportOptions;
```

Parameter Pick Lists

Several new constants, structures, and functions have been added for modifying parameter field pick lists and descriptions.

New Structure:

- PEParameterPickListOptions

New Functions:

- PEGetNthParameterValueDesc
- PEGetNthParameterValueDesc
- PEGetParameterPickListOptions
- PEGetParameterPickListOptions

```

BOOL CRPE_API PEGetNthParameterValueDescription (short
printJob,
const char FAR
* parameterFieldName,
const char FAR * reportName, short index,
HANDLE FAR * valueDesc,
short FAR *valueDescLength);

```

```

BOOL CRPE_API PEGetNthParameterValueDescription (short
printJob,
const char FAR
* parameterFieldName,
const char FAR * reportName, short index,
char FAR * valueDesc);

```

```

// constants for sortMethod in PEParameterPickListOption
struct

```

```

#define PE_OR_NO_SORT                0
#define PE_OR_ALPHANUMERIC_ASCENDING 1
#define PE_OR_ALPHANUMERIC_DESCENDING 2
#define PE_OR_NUMERIC_ASCENDING      3
#define PE_OR_NUMERIC_DESCENDING     4

```

```

typedef struct PEParameterPickListOption

```

```

{
    WORD StructSize;           // initialize to
PE_SIZEOF_PICK_LIST_OPTION
    short showDescOnly;       // boolean value or PE_UNCHANGED
    short sortMethod;         // enum type const, PE_UNCHANGED
for no change
    short sortBasedOnDesc;    // boolean value or PE_UNCHANGED
} PEParameterPickListOption;

#define PE_SIZEOF_PICK_LIST_OPTION (sizeof
(PEParameterPickListOption))

```

```

BOOL CRPE_API PEGetParameterPickListOption (short printJob,
const char FAR * parameterFieldName,
const char FAR * reportName,
PEParameterPickListOption FAR * pickListOption);

BOOL CRPE_API PEsSetParameterPickListOption (short printJob,
const char FAR * parameterFieldName,
const char FAR * reportName, PEParameterPickListOption FAR
* pickListOption);

```

Parameter Range/Discrete Values

The PEParameterValueInfo structure has a new member called hasDiscreteValues which determines whether the parameter accepts discrete values, a range, or both.

New constants and structure members are indicated in bold:

```

// parameter field origin
#define PE_PO_REPORT          0
#define PE_PO_STOREDPROC     1
#define PE_PO_QUERY          2

// range info
#define PE_RI_INCLUDEUPPERBOUND 1
#define PE_RI_INCLUDELOWERBOUND 2
#define PE_RI_NOUPPERBOUND      4
#define PE_RI_NOLOWERBOUND      8

#define PE_DR_HASRANGE        0
#define PE_DR_HASDISCRETE     1
#define PE_DR_HASDISCRETEANDRANGE 2

typedef struct PEParameterValueInfo {
    WORD StructSize;
    short isNullable;           // Boolean value or PE_UNCHANGED for
                                // no change.
    short disallowEditing;      // Boolean value or PE_UNCHANGED for no // change.
    short allowMultipleValues;  // Boolean value or PE_UNCHANGED for no // change.
    short hasDiscreteValues;    // int value or PE_UNCHANGED for no
                                // change.
                                // 0 means has ranges, 1 means has
                                // discrete values
                                // 2 means has discrete
                                and ranged values
    short partOfGroup;          // Boolean value or PE_UNCHANGED for no // change.
    short groupNum;             // a group number or PE_UNCHANGED for no //
                                change.
    short mutuallyExclusiveGroup; // Boolean value or PE_UNCHANGED for no // change.
} PEParameterValueInfo;

#define PE_SIZEOF_PARAMETER_VALUE_INFO (sizeof
(PEParameterValueInfo))

```


New Constants for Time and DateTime values

```
// use these constants for Time and DateTime fields
#define PE_GC_BYSECOND      8
#define PE_GC_BYMINUTE     9
#define PE_GC_BYHOUR       10
#define PE_GC_BYAMPM       11
```

New Events

The PEMouseClickEventInfo structure can be used when a mouse click event has occurred.

New constants and structure members are indicated in bold:

```
// event ID
#define PE_CLOSE_PRINT_WINDOW_EVENT      1
#define PE_ACTIVATE_PRINT_WINDOW_EVENT  2
#define PE_DEACTIVATE_PRINT_WINDOW_EVENT 3
#define PE_PRINT_BUTTON_CLICKED_EVENT   4
#define PE_EXPORT_BUTTON_CLICKED_EVENT  5
#define PE_ZOOM_LEVEL_CHANGING_EVENT     6
#define PE_FIRST_PAGE_BUTTON_CLICKED_EVENT 7
#define PE_PREVIOUS_PAGE_BUTTON_CLICKED_EVENT 8
#define PE_NEXT_PAGE_BUTTON_CLICKED_EVENT 9
#define PE_LAST_PAGE_BUTTON_CLICKED_EVENT 10
#define PE_CANCEL_BUTTON_CLICKED_EVENT  11
#define PE_CLOSE_BUTTON_CLICKED_EVENT   12
#define PE_SEARCH_BUTTON_CLICKED_EVENT  13
#define PE_GROUP_TREE_BUTTON_CLICKED_EVENT 14
#define PE_PRINT_SETUP_BUTTON_CLICKED_EVENT 15
#define PE_REFRESH_BUTTON_CLICKED_EVENT  16
#define PE_SHOW_GROUP_EVENT             17
#define PE_DRILL_ON_GROUP_EVENT          18 // include
drill on graph
#define PE_DRILL_ON_DETAIL_EVENT         19
#define PE_READING_RECORDS_EVENT        20
#define PE_START_EVENT                   21
#define PE_STOP_EVENT                     22
#define PE_MAPPING_FIELD_EVENT           23
#define PE_RIGHT_CLICK_EVENT             24 // right
mouse click
#define PE_LEFT_CLICK_EVENT              25 // left
mouse click
#define PE_MIDDLE_CLICK_EVENT            26 // middle
mouse click

// mouse click action
#define PE_MOUSE_NOTSUPPORTED            0
#define PE_MOUSE_DOWN                    1
#define PE_MOUSE_UP                       2
#define PE_MOUSE_DOUBLE_CLICK            3

typedef struct PEMouseClickEventInfo
{
    WORD StructSize;
    long windowHandle;
```

```

        UINT clickAction;           // one of PE_MOUSE_*
        UINT clickFlags;           // any combination of
PE_CF_*
        int xOffset;               // x-coordinate of
mouse click in pixels
        int yOffset;               // y-coordinate of
mouse click in pixels
PEValueInfo fieldValue;          // value of object at click point if it // is a
                                // field object, excluding MEMO and // BLOB
                                // fields, else valueType element = //
                                PE_VI_NOVALUE.
        DWORD objectHandle;        // the design view
object
        short sectionCode;         // section in which
click occurred.
} PEMouseClickEventInfo;

#define PE_SIZEOF_MOUSE_CLICK_EVENT_INFO (sizeof
(PEMouseClickEventInfo))

```

Graphing

Several new graphing constants, structures, and functions have been introduced.

Structures:

- PEGraphTypeInfo
- PEGraphOptionsInfo
- PEGraphAxisInfo

Functions:

- PEGetGraphTypeInfo
- PEGetGraphOptionsInfo
- PEGetGraphTextInfo
- PEGetGraphFontInfo
- PEGetGraphOptionInfo
- PEGetGraphAxisInfo
- PEGetGraphAxisInfo
- PEGetGraphAxisInfo

```

//graph type
#define PE_GT_BARCHART           0
#define PE_GT_LINECHART         1
#define PE_GT_AREACHART         2

```

```

#define PE_GT_PIECHART 3
#define PE_GT_DOUGHNUTCHART 4
#define PE_GT_THREEDRISERCHART 5
#define PE_GT_THREEDSURFACECHART 6
#define PE_GT_SCATTERCHART 7
#define PE_GT_RADARCHART 8
#define PE_GT_BUBBLECHART 9
#define PE_GT_STOCKCHART 10
#define PE_GT_USERDEFINEDCHART 50 // <----|__ for PEGetGraphTypeInfo // only -
#define PE_GT_UNKNOWNTYPECHART 100 // <----| do not use in
// PEGetGraphTypeInfo.
// PEGetGraphTypeInfo.

// graph subtype
// bar charts
#define PE_GST_SIDEBYSIDEBARCHART 0
#define PE_GST_STACKEDBARCHART 1
#define PE_GST_PERCENTBARCHART 2
#define PE_GST_FAKED3DSIDEBYSIDEBARCHART 3
#define PE_GST_FAKED3DSTACKEDBARCHART 4
#define PE_GST_FAKED3DPERCENTBARCHART 5

// line charts
#define PE_GST_REGULARLINECHART 10
#define PE_GST_STACKEDLINECHART 11
#define PE_GST_PERCENTAGELINECHART 12
#define PE_GST_LINECHARTWITHMARKERS 13
#define PE_GST_STACKEDLINECHARTWITHMARKERS 14
#define PE_GST_PERCENTAGELINECHARTWITHMARKERS 15

//area charts
#define PE_GST_ABSOLUTEAREACHART 20
#define PE_GST_STACKEDAREACHART 21
#define PE_GST_PERCENTAREACHART 22
#define PE_GST_FAKED3DABSOLUTEAREACHART 23
#define PE_GST_FAKED3DSTACKEDAREACHART 24
#define PE_GST_FAKED3DPERCENTAREACHART 25

// pie charts
#define PE_GST_REGULARPIECHART 30
#define PE_GST_FAKED3DREGULARPIECHART 31
#define PE_GST_MULTIPLEPIECHART 32
#define PE_GST_MULTIPLEPROPORTIONALPIECHART 33

```

```

// doughnut charts
#define PE_GST_REGULARDOUGHNUTCHART          40
#define PE_GST_MULTIPLEDOUGHNUTCHART        41
#define PE_GST_MULTIPLEPROPORTIONALDOUGHNUTCHART 42

// 3D riser charts
#define PE_GST_THREEDREGULARCHART           50
#define PE_GST_THREEDPYRAMIDCHART          51
#define PE_GST_THREEDOCTAGONCHART         52
#define PE_GST_THREEDCUTCORNERSCHART       53

// 3D surface charts
#define PE_GST_THREEDSURFACEREGULARCHART    60
#define PE_GST_THREEDSURFACEWITHSIDESCHART 61
#define PE_GST_THREEDSURFACEHONEYCOMBCHART 62

// scatter charts
#define PE_GST_XYSCATTERCHART               70
#define PE_GST_XYSCATTERDUALAXISCHART      71
#define PE_GST_XYSCATTERWITHLABELSCHART    72
#define PE_GST_XYSCATTERDUALAXISWITHLABELSCHART 73

// radar charts
#define PE_GST_REGULARRADARCHART           80
#define PE_GST_STACKEDRADARCHART          81
#define PE_GST_RADARDUALAXISCHART         82

// bubble charts
#define PE_GST_REGULARBUBBLECHART          90
#define PE_GST_DUALAXISBUBBLECHART        91

// stocked charts
#define PE_GST_HIGHLOWCHART                100
#define PE_GST_HIGHLOWDUALAXISCHART       101
#define PE_GST_HIGHLOWOPENCHART           102
#define PE_GST_HIGHLOWOPENDUALAXISCHART   103
#define PE_GST_HIGHLOWOPENCLOSECHART      104
#define PE_GST_HIGHLOWOPENCLOSEDUALAXISCHART 105

#define PE_GST_UNKNOWNSUBTYPECHART        1000

typedef struct PEGraphTypeInfo
{
    WORD StructSize;
    short graphType;      // PE_GT_*, PE_UNCHANGED for no
change
    short graphSubtype;  // PE_GST_*, PE_UNCHANGED for no
change
} PEGraphTypeInfo;

#define PE_SIZEOF_GRAPH_TYPE_INFO (sizeof (PEGraphTypeInfo))

BOOL CRPE_API PEGetGraphTypeInfo(short printJob,
                                short sectionN,
                                short graphN,
                                PEGraphTypeInfo FAR *
graphTypeInfo);

```

```

BOOL CRPE_API PSetGraphTypeInfo(short printJob,
                                short sectionN,
                                short graphN,
                                PEGraphTypeInfo FAR *
graphTypeInfo);

//graph text

// graph text
#define PE_GTT_TITLE           0
#define PE_GTT_SUBTITLE       1
#define PE_GTT_FOOTNOTE       2
#define PE_GTT_SERIESTITLE    3
#define PE_GTT_GROUPSTITLE    4
#define PE_GTT_XAXISTITLE     5
#define PE_GTT_YAXISTITLE     6
#define PE_GTT_ZAXISTITLE     7

// graph text fonts
#define PE_GTF_TITLEFONT      0
#define PE_GTF_SUBTITLEFONT   1
#define PE_GTF_FOOTNOTEFONT   2
#define PE_GTF_GROUPSTITLEFONT 3
#define PE_GTF_DATATITLEFONT  4
#define PE_GTF_LEGENDFONT     5
#define PE_GTF_GROUPLABELSFONT 6
#define PE_GTF_DATALABELSFONT 7

#define PE_FACE_NAME_LEN     64

typedef struct PFontColorInfo
{
    WORD StructSize;
    char faceName[PE_FACE_NAME_LEN];    // empty string for
no change
    short fontFamily;    // FF_DONTCARE for no change
    short fontPitch;    // DEFAULT_PITCH for no change
    short charSet;    // DEFAULT_CHARSET for no change
    short pointSize;    // 0 for no change
    short isItalic;    // BOOL value, except use
PE_UNCHANGED for no change.
    short isUnderlined; // BOOL value, except use
PE_UNCHANGED for no change.
    short isStruckOut; // BOOL value, except use
PE_UNCHANGED for no change.
    short weight;    // 0 for no change
    COLORREF color;    // PE_UNCHANGED_COLOR for no
change.
}PFontColorInfo;

#define PE_SIZEOF_FONT_COLOR_INFO (sizeof(PFontColorInfo))

BOOL CRPE_API PGetGraphTextInfo(short printJob,
                                short sectionN,
                                short graphN,
                                WORD titleType,

//PE_GTT_*

                                HANDLE FAR * title,
                                short FAR * titleLength);

```

```

BOOL CRPE_API PEGraphTextInfo(short printJob,
                              short sectionN,
                              short graphN,
                              WORD titleType,
                              LPCSTR title);

//graph font
BOOL CRPE_API PEGraphFontInfo(short printJob,
                              short sectionN,
                              short graphN,
                              WORD titleFontType,
                              //PE_GTF_
                              PEFontColorInfo FAR *
                              fontColourInfo);

BOOL CRPE_API PEGraphFontInfo(short printJob,
                              short sectionN,
                              short graphN,
                              WORD titleFontType,
                              //PE_GTF_
                              PEFontColorInfo FAR *
                              fontColourInfo);

// graph options
#define PE_GLP_PLACEUPPERRIGHT      0
#define PE_GLP_PLACEBOTTOMCENTER   1
#define PE_GLP_PLACETOPCENTER      2
#define PE_GLP_PLACERIGHT          3
#define PE_GLP_PLACELEFT           4

// bar sizes
#define PE_GBS_MINIMUMBARSIZE      0
#define PE_GBS_SMALLBARSIZE        1
#define PE_GBS_AVERAGEBARSIZE     2
#define PE_GBS_LARGEBARSIZE        3
#define PE_GBS_MAXIMUMBARSIZE     4

// pie sizes
#define PE_GPS_MINIMUMPIESIZE      64
#define PE_GPS_SMALLPIESIZE        48
#define PE_GPS_AVERAGEPIESIZE     32
#define PE_GPS_LARGEPIESIZE        16
#define PE_GPS_MAXIMUMPIESIZE     0

// detached pie slice
#define PE_GDPS_NODETACHMENT        0
#define PE_GDPS_SMALLESTSLICE      1
#define PE_GDPS_LARGESTSLICE        2

// marker sizes
#define PE_GMS_SMALLMARKERS          0
#define PE_GMS_MEDIUMSMALLMARKERS   1
#define PE_GMS_MEDIUMMARKERS         2
#define PE_GMS_MEDIUMLARGEMARKERS   3
#define PE_GMS_LARGEMARKERS          4

// marker shapes
#define PE_GMSP_RECTANGLESHAPE      1
#define PE_GMSP_CIRCLESHAPE         4
#define PE_GMSP_DIAMONDSHAPE        5
#define PE_GMSP_TRIANGLESHAPE       8

```

```

// chart color
#define PE_GCR_COLORCHART 0
#define PE_GCR_BLACKANDWHITECHART 1

// chart data points
#define PE_GDP_NONE 0
#define PE_GDP_SHOWLABEL 1
#define PE_GDP_SHOWVALUE 2

// number formats
#define PE_GNF_NODECIMAL 0
#define PE_GNF_ONEDECIMAL 1
#define PE_GNF_TWODECIMAL 2
#define PE_GNF_CURRENCYNODECIMAL 3
#define PE_GNF_CURRENCYTWODECIMAL 4
#define PE_GNF_PERCENTNODECIMAL 5
#define PE_GNF_PERCENTONEDECIMAL 6
#define PE_GNF_PERCENTTWODECIMAL 7

// viewing angles
#define PE_GVA_STANDARDVIEW 1
#define PE_GVA_TALLVIEW 2
#define PE_GVA_TOPVIEW 3
#define PE_GVA_DISTORTEDVIEW 4
#define PE_GVA_SHORTVIEW 5
#define PE_GVA_GROUPEYEVIEW 6
#define PE_GVA_GROUPEMPHASISVIEW 7
#define PE_GVA_FEWSERIESVIEW 8
#define PE_GVA_FEWGROUPOVIEW 9
#define PE_GVA_DISTORTEDSTDVIEW 10
#define PE_GVA_THICKGROUPOVIEW 11
#define PE_GVA_SHORTERVIEW 12
#define PE_GVA_THICKSERIESVIEW 13
#define PE_GVA_THICKSTDVIEW 14
#define PE_GVA_BIRDSEYEVIEW 15
#define PE_GVA_MAXVIEW 16

typedef struct PEGraphOptionInfo
{
    WORD StructSize;

    short graphColour; // PE_GCR_*, PE_UNCHANGED
for no change

    short showLegend; // BOOL, PE_UNCHANGED for
no change
    short legendPosition; // PE_GLP_*, if showLegend
== 0, means no
// legend

// pie charts and doughnut charts
    short pieSize; // PE_GPS_*, PE_UNCHANGED for no
change
    short detachedPieSlice; // PE_GDPS_*, PE_UNCHANGED
for no change

// bar chart
    short barSize; // PE_GBS_*, PE_UNCHANGED for no
change
    short verticalBars; // BOOL, PE_UNCHANGED for no
change

```

```

// markers (used for line and bar charts)
    short markerSize;           // PE_GMS_*, PE_UNCHANGED
for no change
    short markerShape;        // PE_GMSP_*, PE_UNCHANGED
for no change

//data points
    short dataPoints;         // PE_GDP_*, PE_UNCHANGED
for no change
    short dataValueNumberFormat; // PE_GNF_*, PE_UNCHANGED
for no change

// 3d
    short viewingAngle;      // PE_GVA_*, PE_UNCHANGED for no
change

} PEGraphOptionInfo;

#define PE_SIZEOF_GRAPH_OPTION_INFO (sizeof
(PEGraphOptionInfo))

BOOL CRPE_API PEGetGraphOptionInfo(short printJob,
                                   short sectionN,
                                   short graphN,
                                   PEGraphOptionInfo FAR *
graphOptionInfo);

BOOL CRPE_API PESetGraphOptionInfo(short printJob,
                                   short sectionN,
                                   short graphN,
                                   PEGraphOptionInfo FAR *
graphOptionInfo);

//graph axes
#define PE_GGT_NOGRIDLINES           0
#define PE_GGT_MINORGRIDLINES       1
#define PE_GGT_MAJORGRIDLINES       2
#define PE_GGT_MAJORANDMINORGRIDLINES 3

typedef struct PEGraphAxisInfo
{
    WORD StructSize;

    short groupAxisGridLine;        // PE_GGT_*, PE_UNCHANGED
for no change
    short dataAxisYGridLine;       // PE_GGT_*, PE_UNCHANGED
for no change
    short dataAxisY2GridLine;      // PE_GGT_*, PE_UNCHANGED for
no change
    short seriesAxisGridline;      // PE_GGT_*, PE_UNCHANGED for
no change

    double dataAxisYMinValue;
    double dataAxisYMaxValue;
    double dataAxisY2MinValue;
    double dataAxisY2MaxValue;
    double seriesAxisMinValue;
    double seriesAxisMaxValue;

```



```

    short dataAxisYNumberFormat; // PE_GNF_*, PE_UNCHANGED
for no change
    short dataAxisY2NumberFormat; // PE_GNF_*,
PE_UNCHANGED for no change
    short seriesAxisNumberFormat; // PE_GNF_*,
PE_UNCHANGED for no change

    short dataAxisYAutoRange; // BOOL, PE_UNCHANGED for
no change
    short dataAxisY2AutoRange; // BOOL, PE_UNCHANGED for
no change
    short seriesAxisAutoRange; // BOOL, PE_UNCHANGED for
no change

    short dataAxisYAutomaticDivision; // BOOL,
PE_UNCHANGED for no change
    short dataAxisY2AutomaticDivision; // BOOL,
PE_UNCHANGED for no change
    short seriesAxisAutomaticDivision; // BOOL,
PE_UNCHANGED for no change

long dataAxisYManualDivision;
// if dataAxisYAutomaticDivision is true,
// this field is ignored

long dataAxisY2ManualDivision;
// if dataAxisY2AutomaticDivision is true,
// this field is ignored

long seriesAxisManualDivision;
// if seriesAxisAutomaticDivision is true,
// this field is ignored

} PEGraphAxisInfo;

#define PE_SIZEOF_GRAPH_AXIS_INFO (sizeof
(PEGraphAxisInfo))

BOOL CRPE_API PEGetGraphAxisInfo(short printJob,
                                short sectionN,
                                short graphN,
                                PEGraphAxisInfo FAR *
graphAxisInfo);

BOOL CRPE_API PESetGraphAxisInfo(short printJob,
                                short sectionN,
                                short graphN,
                                PEGraphAxisInfo FAR *
graphAxisInfo);

```

Other Resources

More information on all Crystal API calls can be found in:

- Online Developer's Help installed with Seagate Crystal Reports (developr.hlp)

- Technical Reference Guide:
 - TechRefv1.pdf
 - TechRefv2.pdf
 - TechRefv3.pdf
 - TechRefv4.pdf

These files are located on the Seagate Crystal Reports version 7 MR1 CD

Contacting Crystal Decisions for Technical Support

We recommend that you refer to the product documentation and that you visit our Technical Support web site for more resources.

Self-serve Support:

<http://support.crystaldecisions.com/>

Email Support:

<http://support.crystaldecisions.com/support/answers.asp>

Telephone Support:

<http://www.crystaldecisions.com/contact/support.asp>