

Новое в MapX v4.5.1

В этом документе приводится описание новых возможностей, которые были добавлены к новой версии продукта корпорации MapInfo - MapX v4.5.1. В нижеследующих разделах приводится детальное описание этих возможностей, а также объяснение того, как использовать эти возможности в Ваших приложениях MapX.

- **Совместимость с SpatialWare для Microsoft SQL Server:** Spatial Server Access позволяет разработчикам осуществлять доступ к данным, хранящимся на пространственном сервере предприятия. MapX 4.51 поддерживает работу с новейшим пополнением семейства MapInfo SpatialWare, SpatialWare 4.5 для Microsoft SQL Server.
- **XML Dataset драйвер:** позволяет разработчикам использовать богатые возможности XML. Драйвер MapX XML представляет набор интерфейсов для доступа к данным XML в MapX.
- **Стиль оформления для каждой записи:** Возможность построчного применения стиля к записям в пространственных базах данных, позволяет использовать возможности, которые уже давно заложены в файлах MapInfo TAB. А именно - каждый геометрический объект в одном слое может иметь свой собственный стиль оформления. Это позволяет осуществлять бесперебойное совместное функционирование как отдельных настольных систем, использующих TAB файлы, так и пространственные РСУБД в масштабах предприятия. Следовательно, и пользователи MapX будут видеть одни и те же результаты независимо от того, работают они с TAB файлами или с базой данных, содержащей пространственную информацию. Кроме того, возможность оформить каждую запись освобождает от необходимости разносить объекты по разным слоям и соответственно управлять многочисленными слоями (т.е. теперь в пределах одного слоя один стиль отображения можем использовать для школ, другой для больниц и т.п.).
- **Поддержка ограничения охвата карты (Bounding support):** Используйте **Layer.ClippedBounds** и **Layers.ClippedBounds** для установки свойства Map.Bounds, когда проекция карты имеет бесконечные полюса, например проекция Меркатора, а проекция слоя не имеет. В противном случае, широта должна быть отрегулирована вручную, для того чтобы не получить почти бесконечные числа, что приведет к отображению карты как тонкой вертикальной линии.
- **Улучшенный поиск:** дополнительный необязательный параметр SearchResultFlag был добавлен к методам **Layer.SearchAtPoint** и **Selection.SelectByPoint**. Этот параметр может иметь значение или комбинацию значений из нового набора констант, называемого **SearchResultTypeConstants**. Комбинированием (используя оператор OR) одной или нескольких констант **SearchResultTypeConstants** Вы можете указать, чтобы выбирались все объекты, объекты только верхнего слоя, все регионы, все точки, точки самого верхнего слоя и т.п. По умолчанию возвращаются все объекты. Этот параметр позволяет разработчикам иметь больший контроль над возвращаемым при поиске результатами в слоях, где имеются накладывающиеся объекты или объекты разных типов.

-
- **Свойство Field.TypeEx:** Свойство Field.TypeEx было добавлено в MapX для того, чтобы иметь возможность возвращать ВСЕ имеющиеся типы полей (**FieldTypeConstants**). Раньше свойство **Field.Type** возвращало только String, Date, Numeric.

MapX XML Dataset драйвер

Ниже приводится описание драйвера MapX XML Dataset. Интерфейс MapX Dataset Driver основан на архитектуре объектной модели Microsoft Component Object Model (COM). MapX XML Dataset Driver представляет набор интерфейсов обеспечивающих доступ к данным XML в MapX.

Поддержка XML

Драйвер MapX XML Dataset осуществляет связь с синтаксическим анализатором (Parser) Microsoft XML. Существует несколько версий синтаксического анализатора Microsoft XML. Версия 1.x не поддерживается. Драйвер MapX XML Dataset тестировался и поддерживает версию 2.x. Следующая версия анализатора Microsoft XML 3.0, вероятно, будет поддерживаться, но пока не тестировалась. Другие синтаксические анализаторы в настоящее время не поддерживаются.

Соответствие стандартам драйвера MapX XML Dataset основывается на версии Microsoft XML анализатора зарегистрированного в системе. Например, версия 2.5 синтаксического анализатора Microsoft XML включает следующие соответствия стандартам:

- XML: соответствует стандарту Extensible Markup Language 1.0 версия технических требований.
- XSL: соответствует стандарту XML Stylesheet Language рабочий проект (12.18.1998)

Для получения более детальной информации по синтаксическому анализатору Microsoft XML обратитесь к материалам Microsoft.

Версия 2.0 синтаксического анализатора Microsoft XML включена как часть установки. Однако в момент написания этой документации, версия 2.5 SP1 Microsoft XML стала доступной для самостоятельного распространения и включается как часть установки для Microsoft Windows 2000 SP1 и Microsoft Internet Explorer версии 5.01 или выше.

Помещение данных MapInfo XML Dataset на Вашу карту

Datasets компонента MapX включает метод для добавления нового dataset. Первые два параметра уникальны для каждого драйвера dataset. Первый параметр это идентификатор для драйвера dataset. Каждый драйвер dataset определяется по уникальному числовому коду. Библиотека компонента MapX содержит

предопределенные константы для этих идентификаторов. Предопределенная константа и численный код для драйвера MapX XML Dataset:

```
miDataSetXML = 14
```

Второй параметр - это требуемые для получения данных метаданные для драйвера dataset. В случае драйвера MapX XML Dataset существует два способа передачи необходимых метаданных. Первый и более простой из двух механизмов требует строки, содержащей информацию о размещении MapInfo XML в форме Универсального идентификатора ресурса (uniform resource identifier, URI) или пути (локальный или UNC). В случае указания пути он должен быть относительным к MapXtreme Server. Ниже приведен простой пример на Microsoft Visual Basic:

```
Dim XMLURL As String
Dim XMLDataset As MapXLib.Dataset
'Инициализация XMLURL
XMLURL = "http://www.xml.com/DataSource.xml"
'Предполагается, что MapObject существует на форме
Set XMLDataset = MapObject.Datasets.Add(miDataSetXML, XMLURL)
```

Второй механизм более сложный и включает создание второго объекта автоматизации (automation object) - объекта MXMLDatasetDriverConfiguration. Этот объект имеет два свойства, первое свойство - это SourceLink, представляющее универсальный идентификатор ресурса (URI) или путь указывающий на размещение MapInfo XML Dataset; второе свойство - FilterLink, представляющее URI или путь, к XSLT документу для преобразования указанного XML Dataset, основанного на внешней схеме в MapInfo XML Dataset. Однако возможности XSLT не полностью поддерживаются и отключены в данной реализации драйвера MapX XML Dataset. Следовательно, свойство SourceLink всегда должно указывать на MapInfo XML Dataset. Следующий фрагмент кода - это простой пример для Microsoft Visual Basic (обратите внимание, что установки FilterLink закомментированы):

```
Dim XMLDataset As MapXLib.Dataset
Dim XMLResource As MXMLDatasetDriverConfiguration
' Set XMLResource SourceLink
XMLResource.SourceLink = "http://www.xml.com/DataSource.xml"
' Set XMLResource FilterLink - currently unsupported
' XMLResource.FilterLink = "http://www.xml.com/DataFilter.xml"
' Assume MapObject exists within a form
Set XMLDataset = MapObject.Datasets.Add(miDataSetXML, XMLResource)
```

Создание MapInfo XML Dataset

Под созданием MapInfo XML Dataset подразумевается создание корректного XML документа, разработанного таким способом, чтобы иметь возможность быть обработанным драйвером MapX XML Dataset. Чтобы понять, как это сделать, необходимо представлять, как создать XML документ вообще. Затем, основываясь на этом, рассмотреть схемы, на которых основан MapInfo XML Dataset. Обратите внимание, что MapInfo XML Dataset разрабатывался с учетом схем XML. В данный момент XML схемы не являются универсальным принятым стандартом; эта реализация соответствует возможностям синтаксического анализатора Microsoft XML, версии 2.

Схема, на которой построен корректный MapInfo XML dataset, является подмножеством схемы Microsoft XML rowset. Универсальной имя ресурса (Universal-resource-name) для схемы Microsoft rowset это schemas-microsoft-com:rowset. Нижеследующий фрагмент кода - это базовая схема MapInfo dataset, являющаяся подмножеством схемы Microsoft rowset:

```
<s:Schema xmlns:s='urn:schema-microsoft-com:xml-data'
xmlns:dt='urn:schemas-microsoft-com:datatypes'>
  <!-- baserow definition -->
  <s:ElementType name='baserow' content='eltOnly' />
  <s:element type='baserow' />
  <!-- data definition -->
  <s:ElementType name='data' content='eltOnly'>
  <s:element type='baserow' minOccurs='0' maxOccurs='*' />
  </s:ElementType>
  <s:element type='data' />
</s:Schema>
```

Примечание: В вышеприведенном фрагменте в первой строке идентифицируются два важных пространства имен (names spaces). Первое из двух пространств имен относится к схеме Microsoft schema, которая определяет язык определения схем. Схема определяет структуру допустимого содержания для определенного класса XML документов, или XML документа. Схема MapInfo XML dataset использует встроенную схему, фрагмент документа, определяющий структуру содержания данных. Второе пространство имен относится к схеме Microsoft datatype. Эта схема обеспечивает определения языка для поддерживаемых типов данных.

MapInfo XML Dataset состоит из двух различных фрагментов документа. Первый фрагмент документа - встроенная схема (inline schema). Этот фрагмент схемы следует вышеупомянутой схеме Microsoft schema и определяет структуру MapInfo XML

Dataset. Каждый столбец идентифицирован по имени и типу - однозначным атрибутам элементов строки. Второй фрагмент документа содержит фактические данные. Следуя определению данных схемы MapInfo dataset, фрагмент данных содержит множественные элементы строк, представляющих отдельные строки. В двух нижеследующих примерах показано использование MapInfo XML Dataset, в обоих примерах используются те же самые данные. После рассмотрения двух примеров разобьем XML документ на части и изучим отдельные фрагменты.

Пример А

```
<xml xmlns:s="urn:schema-microsoft-com:xml-data"
xmlns:ds="urn:schemas-microsoft-com:rowset"
xmlns:dt="urn:schemas-microsoft-com:datatypes"
xmlns:x="#DatasetSchema">
  <!-- schema -->
  <s:schema>
    <s:elementType name='row' content='eltOnly'>
      <s:attributeType name='StateAbbr' dt:type='string' />
      <s:attributeType name='StateName' dt:type='string' />
      <s:attributeType name='StatePop' dt:type='long' />
      <extends type='rowbase' />
    </s:elementType>
  </s:schema>
  <!-- data -->
  <ds:data>
    <x:row StateAbbr ='AK' StateName='Alaska' StatePop='550043' />
    <x:row StateAbbr ='NY' StateName='New York' StatePop='17990455' />
  </ds:data>
</xml>
```

Пример В

```
<xml xmlns="#xmldataset">
  <!-- schema -->
  <schema>
    <elementType name='row' content='eltOnly'>
      <attributeType name='StateAbbr' type='string' />
      <attributeType name='StateName' type='string' />
      <attributeType name='StatePop' type='long' />
      <extends type='rowbase' />
    </elementType>
  </schema>
  <data>
    <row StateAbbr ='AK' StateName='Alaska' StatePop=' 550043' />
  </data>
</xml>
```

```
<row StateAbbr ='NY' StateName='New York' StatePop='17990455' />
</data>
</xml>
```

Сначала разберем корневой тэг - <xml> тэг. Важный момент, на который необходимо обратить внимание в этом фрагменте, - это пространство имен (namespaces):

```
<xml xmlns:s="urn:schema-microsoft-com:xml-data"
xmlns:ds="urn:schemas-microsoft-com:rowset"
xmlns:dt="urn:schemas-microsoft-com:datatypes"
xmlns:x="#DatsetSchema">
```

Первые три пространства имен относятся к вышеупомянутой схеме Microsoft schema, 'rowset' и datatypes. Последнее пространство, это внутреннее пространство имен для встроенной схемы. Существует одна важная проблема для обсуждения – использование псевдонимов пространства имен. Драйвер MapX XML Dataset предполагает, что схема Microsoft schema определяет элементы schema, elementType и attributeType. Если эти элементы имеют префикс псевдонима пространства имени, псевдоним должен соотноситься со схемой Microsoft schema. Кроме того, предполагается что атрибут 'type' для определения колонки драйвером MapX XML Dataset определен схемой Microsoft datatypes. Если эти атрибуты имеют префикс псевдонима пространства имени, псевдоним должен соотноситься со схемой Microsoft datatypes. И наконец, предполагается, что элемент data определен в пространстве имен Microsoft rowset. Если элементы данных имеют префикс псевдонима пространства имени, псевдоним должен соотноситься со схемой Microsoft rowset. Драйвер MapX XML Dataset осуществляет явную проверку пространства имен Microsoft, чтобы определить используется ли псевдоним. Если префиксы любого из вышеуказанных элементов ссылаются на неподдерживаемое пространство имен, XML Dataset не будет обработан корректно.

Универсальное имя ресурса (Universal resource name) для пространства имен Microsoft schema - urn:schema-microsoft-com:xml-data. Универсальное имя ресурса для пространства имен Microsoft datatypes - urn:schemas-microsoft-com:datatypes. Универсальное имя ресурса для пространства имен Microsoft rowset - urn:schemas-microsoft-com:rowset.

Посмотрим на *Пример В*. Обратите внимание, только внутреннее пространство имен упомянуто в корневом тэге. Также заметьте, что для элементов, указанных выше, не используются псевдонимы пространства имен. Это будет проверено и обработано драйвером MapX XML. Что касается внутреннего пространства имен, драйвер MapX

XML Dataset не имеет каких либо конкретных требований до тех пор, пока документ является корректным.

Теперь еще раз посмотрим в *Примере А* на фрагмент для встроенной схемы (inline schema):

```
<!-- schema -->
<s:schema>
<s:elementType name='row' content='eltOnly'>
<s:attributeType name='StateAbbr' dt:dt='string' />
<s:attributeType name='StateName' dt:dt='string' />
<s:attributeType name='StatePop' dt:dt='long' />
<extends type='rowbase' />
</s:elementType>
</s:schema>
```

Эта схема идентифицирует элемент называемый строкой (*row*), которая фактически расширяет элемент *rowbase*, элемент определяемый в вышеупомянутой схеме *MapInfo dataset* и схеме *Microsoft rowset*. Согласно встроенной схеме, элемент *row* состоит из трех атрибутов. Эти атрибуты являются колонками строки. Это *StateAbbr* атрибут/колонка типа *string*, *StateName* атрибут/колонка типа *string* и *StatePop* атрибут/колонка типа *long*.

Имеется один важный момент, на который необходимо обратить внимание – тэги XML элементов чувствительны к регистру. Текущая версия драйвера *MapX XML Dataset* предполагает *schema* вместо *Schema*, *elementType* вместо *ElementType*, и *attributeType* вместо *AttributeType*. Если этого не придерживаться, встроенная схема не будет обработана так, как ожидается.

Обращаясь к *Примеру В*, обратите также внимание, что псевдоним пространства имен не используется как префикс для элементов *schema*, *elementType* и *attributeType*, в остальном встроенная схема остается неизменной.

В заключении посмотрим на фрагмент для данных из *Примера А*:

```
<!-- data -->
<ds:data>
<x:row StateAbbr='AK' StateName='Alaska' StatePop='550043' />
<x:row StateAbbr='NY' StateName='New York' StatePop='17990455' />
</ds:data>
```

Ключевой элемент этого фрагмента - элемент данных, который в *Примере А* определен в схеме MapInfo `dataset` и схеме Microsoft `rowset`. Поэтому тэг имеет префикс с соответствующим псевдонимом пространства имен, в данном случае это `ds`. Согласно схемам, он содержит один или более элементов основанных на элементе `rowbase`, который в данном случае является элементом `row`, определенным в вышеупомянутой встроенной схеме. Каждый из этих элементов имеет один атрибут для каждой колонки, идентифицированной во внутренней схеме. Обратите внимание, что элемент `row` имеет префикс в виде псевдонима пространства имен. Еще раз, обращаясь к *Примеру В*, запомните что единственное отличие - это отсутствие псевдонимов пространства имен.

Стиль оформления для каждой записи

Возможность построчного применения стиля к записям в пространственных базах данных дает функциональность, которая уже давно была реализована в файлах MapInfo TAB. А именно, каждый геометрический объект в одном слое может иметь свой собственный стиль оформления. Например, единый слой 'Общественные учреждения' в Oracle8i Spatial может содержать школы, церкви, библиотеки и каждый тип учреждений может быть представлен собственным символом (т.е. один и тот же символ для всех школ). Точно так же, слой дорог в SpatialWare SQL Server может иметь различные условные обозначения для различных классов дорог, например, двойная красная линия для автомобильных дорог 1-го порядка, одиночная красная двухпиксельная линия для главных дорог и тонкая однопиксельная линия для прочих дорог.

Колонка	Тип данных	Описание
RENDITIONTYPE	INTEGER	Численный индикатор указывающий тип оформления записей в таблице. Допустимые значения: <ul style="list-style-type: none">• 0: Для указанной таблицы не работает режим применения стиля оформления к отдельной записи. Объекты будут читаться/обновляться как прежде, используя стиль используемый для таблицы по умолчанию.• 1: Указанная таблица использует режим применения стиля оформления для каждой записи. Стили оформления записаны в колонке RENDITIONCOLUMN как строка MapBasic (т.е. тот же формат который используется в каталоге карт (Map Catalog) в колонке SYMBOL).
RENDITIONCOLUMN	VARCHAR(32)	Эта колонка содержит имя колонки указанной таблицы, в которой содержится значение стиля объекта (для типа оформления 1).

Колонка	Тип данных	Описание
RENDITIONTABLE	VARCHAR(32)	В настоящее время эта колонка не используется, однако, должна быть представлена до того как MapX будет иметь возможность использовать эту новую функциональность. В будущем предполагается управлять стилями оформления объектов помещая стили оформления в отдельную таблицу, а в этой колонке иметь ключевое значение указывающее на стиль оформления в таблице стилей. Эта функциональность пока не реализована.

Примечание: Если эти колонки не представлены, стиль объектов будет работать как раньше (т.е. стиль оформления таблицы будет применен ко всем объектам таблицы) с одним исключением, обсуждаемым ниже. Если эти колонки представлены в Каталоге карт (Map Catalog), они будут использованы следующим образом:

Известные ограничения/требования

При использовании слоев, которые запрашивают столбец стилей, Вы можете встретиться с проблемой обновления существующих объектов (или значений данных строки) и вставкой новых строк. Это происходит потому, что MapX автоматически добавляет колонку стиля при использовании предложения INSERT/UPDATE (вставить/обновить) по мере необходимости, и может для слоев, не имеющих скрытой колонки стиля, сгенерировать SQL выражение, которое укажет имя колонки дважды.

Свойство `Layers.ClippedBounds` (Коллекция `Layer`)

Назначение

Это свойство, только для чтения возвращает объединение всех слоев, усеченных границами объекта прямоугольник (`Rectangle object`). Используйте это свойство, чтобы установить свойство `Map.Bounds`, когда карта имеет проекцию с бесконечными полюсами, такую как проекция Меркатора, а проекция слоя нет. В противном случае широта должна быть отрегулирована вручную для того, чтобы не получить почти бесконечные числа (что приведет к отображению карты как тонкой вертикальной линии).

Примечание

Поскольку это свойство только для чтения, Вы НЕ можете задать другой прямоугольный объект.

Пример

```
Set Map.Bounds = Map.Layers.ClippedBounds
```

Смотрите также

Свойство `Map.Bounds`

Свойство `Layer.Bounds`

Свойство `Layer.ClippedBounds`

Примечание: Обратитесь к документации по `MapX` за дополнительной информацией об использовании перечисленных выше свойств.

Свойство `Layer.ClippedBounds` (Объект `Layer`)

Назначение

Это свойство, только для чтения, возвращает объединение всех слоев, усеченных границами объекта прямоугольник (`Rectangle object`). Используйте это свойство, чтобы установить свойство `Map.Bounds`, когда карта имеет проекцию с бесконечными полюсами, такую как проекция Меркатора, а проекция слоя нет. В противном случае широта должна быть отрегулирована вручную для того, чтобы не получить почти бесконечные числа (что приведет к отображению карты как тонкой вертикальной линии).

Примечание

Поскольку это свойство только для чтения, Вы НЕ можете задать другой прямоугольный объект.

Пример

```
Set Map.Bounds = Map.Layer.ClippedBounds
```

Смотрите также

Свойство `Map.Bounds`

Свойство `Layer.Bounds`

`Layers.ClippedBounds`

Примечание: Для того, чтобы получить дополнительную информацию по вышеперечисленным свойствам, смотрите документацию по `MapX`.

Метод `Layer.SearchAtPoint` (Объект `Layer`)

Назначение

Возвращает коллекцию `Features`.

- Регионы - если точка лежит внутри полигона.
- Линии и символы - ближайшая точка или линия (может быть несколько, если точки имеют одинаковые координаты) в пределах 3 пиксел вокруг заданной точки будут добавлены к результирующей коллекции.
- Набор объектов - комбинированием (используя оператор `OR`) одной или нескольких констант **`SearchResultTypeConstants`** Вы можете указать, чтобы выбирались все объекты, объекты только верхнего слоя, все регионы, все точки, точки самого верхнего слоя и т.п. По умолчанию возвращаются все объекты.

Синтаксис

[`Features collection`]=`OBJECT.SearchAtPoint` (`Point`, [`SearchResultFlag`])

Часть	Описание
<code>OBJECT</code>	Объект <code>Layer</code> .
<code>Point</code>	координаты (X , Y) задаваемой точки на карте.
<code>SearchResultFlag</code>	Дополнительный необязательный параметр, контролирующий данные, возвращаемые в результате запроса. Определяется одним или несколькими значениями <code>SearchResultTypeConstants</code>

`SearchResultTypeConstants`

`miSearchResultDefault` = 0x0000

`miSearchResultTopmost` = 0x0001

`miSearchResultRegion` = 0x0002

`miSearchResultPoint` = 0x0004

`miSearchResultLine` = 0x0008

`miSearchResultText` = 0x0010

`miSearchResultAll` = 0xFFFFE

Пример на VB (использование SearchResultFlag)

```

Dim aPrivate Sub Map1_MouseDown(Button As Integer, Shift As
Integer, X As Single, Y As Single)
    Dim Lon As Double
    Dim lat As Double
    Dim fs As Features
    Dim pnt As New Point
    If Map1.CurrentTool = miArrowTool Then
        Map1.ConvertCoord X, Y, Lon, lat, miScreenToMap
        pnt.Set Lon, lat
        Set fs = Map1.Layers("USA").SearchAtPoint(pnt,
miSearchResultTopmost OR miSearchResultRegion)
        If fs.Count > 0 Then
            MsgBox fs.Count & ": " & fs(1).Name
        Else
            MsgBox "Ничего не найдено"
        End If
    End If
End Sub

```

Пример на C++ (использование SearchResultFlag)

```

CMapXFeatures CSampleClass::SelectTopmostRegion(const
CMapXPoint& point)
{
    CMapXFeatures fs;
    try
    {
        fs = m_oMap.GetLayers().Item("USA").SearchAtPoint(point,
miSearchResultTopmost | miSearchResultRegion);
    }
    catch()
    {
        e->ReportError();
        e->Delete();
    }
    return fs;
}

```

Метод **Selection.SelectByPoint** (Коллекция Selection)

Назначение

Этот метод позволяет выбрать объект(ы) в определенной точке слоя.

Синтаксис

OBJECT.**SelectByPoint** (X, Y, SelectFlag, [SearchResultFlag])

Часть	Описание
OBJECT	Представляет коллекцию Selection.
X	Координата X точки в которой осуществляется выбор. Значение двойной точности (Долгота).
Y	Координата Y точки в которой осуществляется выбор. Значение двойной точности (Широта).
SelectFlag	Этот параметр определяет, будут ли выбранные объекты добавляться, удаляться или замещаться в текущей выборке. Принимает значение SelectionTypeConstants .
SearchResultFlag	Необязательный параметр, контролирующий, что должно возвращаться в результате запроса. Принимает одно или комбинацию значений SearchResultTypeConstants .

Примечание

Если Вы выбираете полигон, то метод Selection.SelectByPoint выберет полигон в который попадает точка запроса. Если Вы выбираете точки или линии, метод будет выбирать точки и линии в пределах 3 пиксел от точки запроса.

Комбинированием (используя оператор OR) одной или нескольких констант **SearchResultTypeConstants** Вы можете указать чтобы выбирались все объекты, объекты только верхнего слоя, все регионы, все точки, точки самого верхнего слоя и т.п. По умолчанию возвращаются все объекты.

SearchResultTypeConstants

miSearchResultDefault = 0x0000

Метод Selection.SelectByPoint (Коллекция Selection)

miSearchResultTopmost = 0x0001

miSearchResultRegion = 0x0002

miSearchResultPoint = 0x0004

miSearchResultLine = 0x0008

miSearchResultText = 0x0010

miSearchResultAll = 0xFFFF

Свойство `Field.TypeEx` (Объект `Field`)

Назначение

Это свойство возвращает тип поля со значением одной из констант

`FieldTypeConstants`. Это свойство только для чтения.

Примечание: Это новое для MapX свойство. Оно было реализовано для исправления проблемы со свойством `Field.Type`. Свойство `Field.Type` может возвращать только типы `String`, `Date`, и `Numeric`. Свойство `Field.TypeEx` может возвращать ВСЕ типы (`FieldTypeConstants`).

Примечание

В таблице приведено описание возвращаемых типов данных при использовании `Field.Type` и `Field.TypeEx`.

Тип поля	Значение, возвращаемое при использовании <code>Field.Type</code>	Значение, возвращаемое при использовании <code>Field.TypeEx</code>
Integer	numeric	Integer
Character	Character	Character
Float	numeric	Float
Logical	numeric	Logical
Date	Date	Date
Small Int	numeric	Small Int
Decimal(4,2)	numeric	numeric

Смотрите также

Свойство `Field.Type`

`FieldTypeConstants`

`miTypeString` = 0

`miTypeNumeric` = 1

`miTypeDate` = 2

`miTypeInt` = 3

miTypeSmallInt = 4

miTypeFloat = 5

miTypeLogical = 6

