

Краткий справочник по Microsoft Dynamics NAV 5.0

Содержание:

AUTOMATION	2	INSTREAM & OUTSTREAM.....	18
BLOB	2	KEYREF	18
CODEUNIT	3	NUMERIC.....	19
CONTROLS	4	RECORD.....	19
DATAPORT	7	RECORDID	25
DATES & TIMES.....	8	RECORDREF	26
DIALOG	10	REPORT	30
ERROR HANDLING.....	11	STRINGS	33
FIELDREF.....	11	SYSTEM.....	34
FILE	13	VARIANT.....	38
FORM.....	15	VIRTUAL TABLES.....	39
GUID.....	17	XMLPORT	39

AUTOMATION

Тип данных «Automation» (автоматизация) используется для работы с сервером автоматизации. Чтобы использовать сервер автоматизации в C/SIDE, определите тип переменной Automation и задайте ей имя. C/SIDE может принимать события от сервера автоматизации.

CREATE	[Ok] := CREATE(Automation [,NewServer]) Используется для создания объекта автоматизации.
ISCLEAR	Ok := ISCLEAR(Automation) Используется для того, чтобы проверить, был ли создан объект автоматизации.
VARIABLEACTIVE	IsActive := VARIABLEACTIVE(Variable) Позволяет определить, активны или нет такие переменные, как поле или элемент управления.

BLOB

BLOBs (Binary Large Object, большой двоичный объект) имеет переменную длину. Максимальный размер переменной BLOB обычно определяется размером жесткого диска. Однако максимальный размер в C/SIDE составляет 2ГБ. Примечание: C/SIDE не может отобразить текст, хранимый в BLOB.

CREATEINSTREAM	Blob.CreateInStream(Stream) Создает объект InStream (входящий поток) для BLOB. Позволяет преобразовать данные в тип BLOB.
CREATEOUTSTREAM	Blob.CreateOutStream(Stream) Создает объект OutStream (исходящий поток) из BLOB. Позволяет извлечь данные из BLOB.
EXPORT	[ExportName] := Blob.EXPORT([Name [, CommonDialog]]) Используется для экспорта BLOB.
HASVALUE	HasValue := Blob.HASVALUE Определяет, есть ли какое-либо значение у объекта типа BLOB.
IMPORT	[ImportName] := Blob.IMPORT([Name [, CommonDialog]]) Используется для импорта BLOB.

CODEUNIT

Данный комплексный тип данных используется для хранения элементов кода (codeunit) в C/AL. Codeunit'ы содержат количество определенных пользователем функций.

Codeunit.RUN [Ok] := Codeunit.RUN(Number [, Record])

Используется для загрузки и выполнения codeunit'а в C/AL.

Если вы не включите возвращаемое значение (опционально), и система выдаст сообщение об ошибке, выполнение C/AL кода прекратится.

Если вы включите возвращаемое значение, и система выдаст сообщение об ошибке, она продолжит выполнение кода C/AL. Но в дальнейшем вы должны устранить все эти ошибки.

RUN [Ok] := Codeunit.RUN(VAR Record)

Используется для загрузки и выполнения codeunit'а в C/AL. Чтобы использовать эту функцию, вы можете определить таблицу C/SIDE, связанную с codeunit'ом. Это даст возможность передать переменную с функцией. Транзакция, которая содержится в codeunit'е, всегда передается в виде булевого возвращаемого значения.

CONTROLS

ACTIVATE	[Ok] := Form.ACTIVATE Используйте эту функцию, чтобы сделать форму или элемент управления активным.
DECIMALPLACESMAX	[CurrMaxDecimals] :=DECIMALPLACESMAX([NewMaxDecimals]) Используется для возврата текущей настройки максимального числа символов в элементе управления (поле или textbox), и для установки нового значения.
DECIMALPLACESMIN	[CurrMinDecimals] := DECIMALPLACESMIN([NewMinDecimals]) Используется для возврата текущей настройки минимального числа символов в элементе управления (поле или textbox), и для установки нового значения.
EDITABLE	[IsEditable] := Form.EDITABLE([SetEditable]) Используется для возврата текущей настройки свойства Editable (редактируемый), и для изменения этого свойства.
ENABLED	[IsEnabled] := ENABLED([SetEnabled]) Используется для возврата текущей настройки свойства Enabled (Включен) элемента управления, и для изменения этого свойства.
HEIGHT	[CurrHeight] := Form.HEIGHT([NewHeight]) Используется для возврата текущей настройки свойства Height (Высота) формы или элемента управления, и для изменения этого свойства.
INLINEEDITING	[IsInLineEditing] := INLINEEDITING([SetInLineEditing]) Используется для возврата текущей настройки свойства InLineEditing для table box или matrix box, и для изменения этого свойства. По умолчанию свойство принимает значение «Yes», и при открытой форме редактирование не будет разрешено. Пользователь должен щелкнуть клавишей мыши на активный редактируемый элемент управления и нажать F2, или вставить новую запись (для дальнейшего редактирования).
UPDATEEDITABLE	UPDATEEDITABLE(Editable) Используется для динамического изменения свойства Editable поля, формы или элемента управления. Эта функция может быть вызвана только из свойства OnBeforeInput элемента управления.
UPDATESELECTED	UPDATESELECTED(Selected) Используется для изменения свойства Selected (выбран) для элемента управления (он будет изменять свой внешний вид, но это зависит от схемы цветов Windows, которую выбрал конечный пользователь). Эта функция может использоваться только в триггере OnFormat.
UPDATEFONTBOLD	UPDATEFONTBOLD(FontBold) Используется для динамического изменения свойства FontBold (жирный шрифт) элемента управления. Эта функция может быть вызвана только из триггера OnFormat.

Microsoft Dynamics NAV 5.0 Quick Reference RU

UPDATEFORECOLOR	<p>UPDATEFORECOLOR(ForeColor)</p> <p>Используется для динамического изменения свойства ForeColor (передний фон) элемента управления. Эта функция может быть вызвана только из триггера OnFormat.</p>
UPDATEINDENT	<p>UPDATEINDENT(Indent)</p> <p>Используется для установки свойства Indent (глубина) у textbox'а. Свойство Indent может быть установлено только при вызове этой функции из триггера OnFormat на textbox'е. Точность - 1/100 мм.</p>
VISIBLE	<p>[IsVisible] := Form.VISIBLE([SetVisible])</p> <p>Используется для возврата текущего значения свойства Visible (видимый) для формы или элемента управления, и для изменения этого свойства.</p>
WIDTH	<p>[CurrWidth] := Form.WIDTH([NewWidth])</p> <p>Используется для возврата текущего значения свойства Width (ширина) для формы или элемента управления, и для установки этого свойства в новое значение.</p>
XPOS	<p>[CurrXPos] := Form.XPOS([NewXPos])</p> <p>Используется для возврата текущего значения свойства XPos для формы или элемента управления, и для установки этого свойства в новое значение. Это свойство определяет горизонтальную позицию формы или элемента управления на экране. Система показывает позицию с точностью до 1/100 миллиметра.</p>
YPOS	<p>[CurrYPos] := Form.YPOS([NewYPos])</p> <p>Используется для возврата текущего значения свойства YPos для формы или элемента управления, и для установки этого свойства в новое значение. Это свойство определяет вертикальную позицию формы или элемента управления. Система показывает позицию с точностью до 1/100 миллиметра.</p>
CHECKLICENSEFILE	<p>CHECKLICENSEFILE(KeyNumber)</p> <p>Используется для проверки ключа файла лицензии в системе.</p>
COMMIT	<p>COMMIT</p> <p>Используется для завершения процесса транзакции записи. При прерывании транзакций все выполненные операции сохраняются. Если вы хотите, чтобы codeunit C/AL выполнял одновременно несколько транзакций записи, вы должны сначала использовать функцию COMMIT для завершения текущей транзакции до того, как начнете следующую. Функция COMMIT разделяет транзакции записи в модуле кода C/AL.</p>
COMPANYNAME	<p>Name := COMPANYNAME</p> <p>Используется для возврата текущего названия фирмы.</p>
CURRENTTRANSACTIONTYPE	<p>[TransactionType] := CURRENTTRANSACTIONTYPE([TransactionType])</p> <p>Используется для возврата текущего типа транзакции и назначения нового. Доступны следующие базовые типы транзакций: Browse, Snapshot, UpdateNoLocks, Update, Report. [SQL]</p>

Microsoft Dynamics NAV 5.0 Quick Reference RU

LOCKTIMEOUT	<p>[LockTimeout] := LOCKTIMEOUT([LockTimeout])</p> <p>Используется для того, чтобы установить значение блокировки при простое. Функция специально была разработана для длительных процессов обработки данных, которые не должны прекращать работу, например ночные задания по планировщику.</p>
SELECTLATESTVERSION	<p>SELECTLATESTVERSION</p> <p>Используется для вызова последней версии БД для использования.</p>
SERIALNUMBER	<p>String := SERIALNUMBER</p> <p>Используется для возврата строки, содержащей серийный номер файла лицензии для вашей системы Microsoft Dynamics NAV.</p>
USERID	<p>ID := USERID</p> <p>Используется для возврата системой ID (идентификационного номера) текущего пользователя.</p>

DATAPOINT

Объекты *Dataport* используются для импорта и экспорта данных во внешние текстовые файлы.

BREAK	<p>BREAK</p> <p>Используется для выхода из цикла или триггера в dataport'ах, отчетах (reports) или XMLport'ах.</p>
DATAPORT.RUN	<p>DATAPORT.RUN(Number [, ReqWindow] [, Record])</p> <p>Используется для загрузки и запуска dataport'а.</p>
DATAPORT.RUNMODAL	<p>DATAPORT.RUNMODAL(Number [, ReqWindow] [, Record])</p> <p>Используется для загрузки и запуска dataport'а.</p>
FILENAME	<p>[CurrFileName] := FILENAME([NewFileName])</p> <p>Используется для возврата текущего значения свойства FileName (имя файла) dataport'а и установки свойства в новое значение. Если FileName оставлен пустым, то будет создана форма запроса по умолчанию.</p>
IMPORT	<p>[IsImport] := IMPORT([SetImport])</p> <p>Используется для возврата текущего значения свойства Import и изменения этого свойства. Импорт может быть установлен только динамически в триггере OnPreDataPort. Если Import оставлен пустым, то будет создана форма запроса по умолчанию.</p>
QUIT	<p>QUIT</p> <p>Используется для прекращения работы dataport'а, report'а или XMLport'а. При использовании функции QUIT dataport, report или XMLport остаются без каких-либо изменений, которые применяются к ним во время выполнения. Триггеры OnPostReport, OnPostDataPort или OnPostXMLport вызываться не будут.</p>
RUN	<p>Dataport.RUN</p> <p>Используется для загрузки и выполнения dataport'а. Система автоматически очищает переменные после выполнения этой функции.</p>
RUNMODAL	<p>Dataport.RUNMODAL</p> <p>Используется для загрузки и выполнения dataport'а. После выполнения этой функции система не очищает переменные автоматически. Необходимо сделать это вручную.</p>
SETTABLEVIEW	<p>SETTABLEVIEW(Record)</p> <p>Используется для применения представления таблицы (Table View) на текущую запись как представление таблицы для формы, отчета или dataport'а. Если при использовании функции свойство формы SaveTableView установлено как "No", система не будет сохранять текущее представление таблицы.</p>
SKIP	<p>SKIP</p> <p>Используется для пропуска (перешагивания) текущей итерации текущего dataport'а, report'а или XMLport'а.</p>

DATES & TIMES

Простой тип данных *DATE* используется для обозначения дат в промежутке от 1го января 0г. (нулевой год) до 31 декабря 9999г. Система обозначает неопределенную дату как *OD*. Тип данных *DATETIME* используется для обозначения даты и времени дня. *Datetime* хранится в БД как *Coordinated Universal Time* (универсальное координатное время, *UTC*). Тип данных *DURATION* (продолжительность) используется для представления разницы между двумя *datetime* в миллисекундах. Это значение может быть отрицательным. Тип – 64битный *integer*. Простой тип данных *TIME* используется для обозначения времени. Система обозначает неопределенное время как *OT*. Любое время в промежутке от 00:00:00 до 23:59:59 действительно. Неопределенная дата в *Microsoft Dynamics NAV* представляется в виде самой ранней действительной даты в сервере *SQL*: 01-01-1753 00:00:00:000 для типа данных *DATETIME*, и 01-01-1900 00:00:00:000 для типа данных *SMALLDATETIME*.

CALCDATE	NewDate := CALCDATE(DateExpression [, Date]) Вычисляет новую дату, основываясь на дате выражения и дате ссылки. Если формула даты введена с разделителями < >, она будет храниться в родном безъязыковом зависимом формате. Это дает возможность разрабатывать формулы для дат, независимых от текущего выбранного языка.
CLOSINGDATE	ClosingDate := CLOSINGDATE(Date) Используется для возврата ближайшей даты для типа <i>Date</i> . Все даты имеют соответствующую ближайшую дату. Ближайшая дата для системы – это та, в которой период следует за заданной датой, но не достигает следующей.
CREATEDATETIME	DateTime := CREATEDATETIME(Date, Time) Используется для создания <i>datetime</i> из даты и времени.
CURRENTDATETIME	Datetime := CURRENTDATETIME Возвращает текущую дату и время.
DATE2DMY	Number := DATE2DMY(Date, What) Возвращает день, месяц или год, основываясь на дате. Параметр "What" определяет, что должно возвращаться: 1 = День, 2 = Месяц, 3 = Год.
DATE2DWY	Number := DATE2DWY(Date, What) Возвращает день недели, номер недели и год, основываясь на параметре <i>Date</i> . Параметр "What" определяет, что должно возвращаться: 1 = День недели, 2 = Номер недели, 3 = Год.
DAT2VARIANT	Variant := DAT2VARIANT(Date, Time) Данная функция системной даты используется для создания <i>variant</i> , содержащей <i>VT_DATE</i> .
DMY2DATE	Date := DMY2DATE(Day [, Month] [, Year]) Используется для возврата даты, основываясь на указанных параметрах <i>day</i> , <i>month</i> , и <i>year</i> .
DT2DATE	Date := DT2DATE(Datetime) Используется для возврата даты из <i>datetime</i> .
DT2TIME	Time := DT2TIME(Datetime) Используется для возврата времени из <i>datetime</i> .

Microsoft Dynamics NAV 5.0 Quick Reference RU

DWY2DATE	Date := DWY2DATE(WeekDay [, Week] [, Year]) Используется для возврата даты, основываясь на дне недели, номере недели и года.
NORMALDATE	NormalDate := NORMALDATE(Date) Используется для возврата нормальной даты (как отклонение от ближайшей даты) для аргумента Date.
ROUND DATETIME	NewDateTime := ROUND DATETIME(Datetime [, Precision][, Direction]) Используется для округления времени и даты.
TIME	Time := TIME Используется для получения текущего времени из ОС.
TODAY	Date := TODAY Используется для возврата текущей даты, установленной в ОС.
VARIANT2DATE	Date := VARIANT2DATE(Variant) Данная функция системной даты используется для возврата даты из VT_DATE.
VARIANT2TIME	Time := VARIANT2TIME(Variant) Данная функция системной даты используется для возврата времени из VT_DATE
WORKDATE	[WorkDate] := WORKDATE([NewDate]) Используется для возврата текущей рабочей даты или установки новой рабочей даты.

DIALOG

Переменные это комплексного типа данных хранят диалоговые окна. Эти переменные также дают доступ к диалоговым функциям, таким, как OPEN, CLOSE и др.

BEEP	<p>BEEP(Frequency, Duration) Позволяет издать звуковой тон из компьютера. Продолжительность - в миллисекундах.</p>
CLOSE	<p>Dialog.CLOSE Используется для закрытия диалогового окна, открытого с помощью функции OPEN.</p>
CONFIRM	<p>Ok := CONFIRM(String [, Default] [, Value1] ,...) Используется для создания диалогового окна, в котором пользователю предлагается отвечать «да» или «нет». Используйте обратный слэш (\) в параметре "String" для перехода на новую строку.</p>
ERROR	<p>ERROR(String [, Value1, ...]) Используется для вывода на экран сообщений об ошибке и завершения выполнения кода C/AL. Используйте обратный слэш (\) в параметре "String" для перехода на новую строку. Используйте знаки процента (%) или знаки # для вставки значений переменных в строку.</p>
INPUT	<p>NewControlID := Dialog.INPUT([ControlID] [,Variable]) Используется для определения того, что вводит пользователь в поле окна. ID – идентификатор следующего поля в окне. Если пользователь нажмет Esc, чтобы покинуть окно, то оно закроется и вызов кода C/AL прекратится, возвращаемым значением будет 0.</p>
MESSAGE	<p>MESSAGE(String [, Value1, ...]) Используется для вывода на экран текстовой строки в окне сообщения. Используйте обратный слэш (\) в параметре "String" для перехода на новую строку. Используйте знаки процента (%) для вставки значений переменных в строку.</p>
OPEN	<p>Dialog.OPEN(String [, Variable1], ...) Открывает диалоговое окно. String содержит текст, который вы хотите видеть в окне. Используйте обратный слэш (\) в параметре "String" для перехода на новую строку. Используйте знаки # для вставки значений переменных в строку. Если вы используете знак @ вместо #, то строку можно будет использовать как индикатор.</p>
STRMENU	<p>OptionNumber := STRMENU(OptionString [, DefaultNumber]) Используется для создания меню, которое показывает возможные опции. Строка OptionString разделяется запятыми. Каждая подстрока в OptionString обозначает конкретную опцию в меню.</p>
UPDATE	<p>Dialog.UPDATE([Number] [, Value]) Используется для обновления значения поля '#' или '@' в текущем окне. Каждое поле '#' или '@' имеет специальный номер. Параметр Number сообщает системе, в какое поле он должен вставить значение.</p>
YIELD	<p>YIELD Используется для передачи контроля операционной системе (на DOS/Windows 3.x), чтобы он мог обрабатывать события. Когда ОС заканчивает обработку, контроль возвращается к пользователю.</p>

ERROR HANDLING

Вы можете настроить Microsoft Dynamics NAV таким образом, чтобы он продолжал работу, даже если произойдет ошибка во время запуска. Вы можете использовать функцию `GETLASTERRORTEXT`, чтобы узнать, была ли ошибка и для того, чтобы посмотреть текст в последнем сгенерированном сообщении об ошибке. Затем вы можете использовать функцию `CLEARLASTERROR` для удаления последнего сообщения об ошибке из памяти.

<code>CLEARLASTERROR</code> [5.0]	<code>CLEARLASTERROR</code> Удаляет последнее сообщение об ошибке из памяти.
<code>GETLASTERRORTEXT</code> [5.0]	<code>String := GETLASTERRORTEXT</code> Используется для возврата текста, содержащегося в последнем сообщении об ошибке. Строка не может превышать максимальную длину строки, определенной в Microsoft Dynamics NAV. На текущий момент это значение - 1024.

FIELDREF

Этот комплексный тип данных идентифицирует поле в таблице и дает к нему доступ. Объект `fieldref` (индекс поля) может ссылаться на любое поле любой таблицы БД.

<code>ACTIVE</code>	<code>Ok := FieldRef.ACTIVE</code> Используется для проверки активности текущего выбранного поля.
<code>CALCFIELD</code>	<code>[Ok] := FieldRef.CALCFIELD</code> Используется для обновления <code>FlowField</code> в записи.
<code>CALCSUM</code>	<code>[Ok] := FieldRef.CALCSUM</code> Используется для вычисления общего <code>SumIndexField</code> в таблице.
<code>CAPTION</code>	<code>Caption := FieldRef.CAPTION</code> Используется для возврата текущего заголовка (<code>caption</code>) поля, получаемого от <code>fieldref</code> как строка. <code>CAPTION</code> первым делом смотрит на свойство <code>CaptionML</code> . В случае его отсутствия используется свойство <code>Name</code> .
<code>CLASS</code>	<code>Class := FieldRef.CLASS</code> Используется для возврата текущего выбранного класса поля.
<code>FIELDERROR</code>	<code>FieldRef.FIELDERROR([Text])</code> Используется для прекращения выполнения кода (как правило при ошибке запуска) и создания сообщения об ошибке для поля.
<code>GETFILTER</code>	<code>String := FieldRef.GETFILTER</code> Используется для возврата фильтра вместе с текущей группой фильтра, примененной к полю.
<code>GETRANGEMAX</code>	<code>Value := FieldRef.GETRANGEMAX</code> Используется для возврата максимального значения диапазона для поля. Тип данных "Value" должен соответствовать типу данных поля, получаемого от <code>fieldref</code> .
<code>GETRANGEMIN</code>	<code>Value := FieldRef.GETRANGEMIN</code> Используется для возврата минимального значения диапазона для поля. Тип данных "Value" должен соответствовать типу данных поля, получаемого от <code>fieldref</code> .

Microsoft Dynamics NAV 5.0 Quick Reference RU

LENGTH	Length := FieldRef.LENGTH Используется для возврата максимального размера поля (размер определяется в свойстве DataLength у поля).
NAME	Name := FieldRef.NAME Используется для возврата имени поля в виде строки.
NUMBER	No := FieldRef.NUMBER Используется для возврата номера поля.
OPTIONCAPTION	OptionCaption := FieldRef.OPTIONCAPTION Используется для возврата заголовка опции текущего выбранного поля.
OPTIONSTRING	OptionString := FieldRef.OPTIONSTRING Используется для возврата списка опций, доступных в текущем выбранном поле.
RECORD	RecordRef := FieldRef.RECORD Используется для возврата recordref текущего выбранного поля.
RELATION	TableNumber := FieldRef.RELATION Используется для поиска связи в таблицах заданного поля.
SETFILTER	FieldRef.SETFILTER(String [, Value],...) Используется для наложения фильтра на поле.
SETRANGE	FieldRef.SETRANGE([FromValue] [, ToValue]) Используется для наложения простого фильтра на поле, такого как простой диапазон или простое значение.
TESTFIELD	FieldRef.TESTFIELD([Value]) Позволяет посмотреть, соответствует ли содержимое поля заданному значению. Тип данных "Value" должен соответствовать типу данных поля.
TYPE	Type := FieldRef.TYPE Используется для возврата текущего выбранного типа данных поля.
VALIDATE	FieldRef.VALIDATE([NewValue]) Используется для ввода нового значения в поле и получения нового значения, получаемого свойствами и кодом, определенными для поля. Параметр "NewValue" должен быть совместим с типом данных поля, получаемого от fieldref.
VALUE	[CurrValue] := FieldRef.VALUE([NewValue]) Используется для установки или получения значения текущего выбранного поля.

FILE

Переменные данного типа данных дают доступ к файлам. Файлы можно открыть в текстовом или двоичном виде.

CLOSE	File.CLOSE Используется для закрытия файла, открытого функцией OPEN.
COPY	[Ok] := File.COPY(FromName, ToName) Используется для копирования файла.
CREATE	[Ok] := File.CREATE(Name) Используется для создания и открытия ASCII или двоичного файла. Если файл существует, система его урежет и затем откроет. Необходимо всегда вызывать функцию TEXTMODE до использования CREATE или OPEN.
CREATEINSTREAM	File.CREATEINSTREAM(Stream) Используется для создания входящего потокового объекта InStream для файла. Это дает возможность пересылать данные в файл.
CREATEOUTSTREAM	File.CREATEOUTSTREAM(Stream) Используется для создания исходящего потокового объекта OutStream для файла. Это дает возможность извлечь данные из файла.
CREATETEMPFILE	File.CREATETEMPFILE Используется для создания временного файла. Это дает возможность хранить данные любого формата во временном файле. Этот файл имеет уникальное имя и хранится в папке временных файлов. При вызове функции CLOSE файл удалится.
ERASE	[Ok] := File.ERASE(Name) Используется для удаления файла.
EXISTS	[Ok] := File.EXISTS(Name) Используется для определения существования файла.
GETSTAMP	[Ok] := File.GETSTAMP(Name, Date [, Time]) Используется для определения последнего времени записи в файл (возвращает метку времени).
LEN	Length := File.LEN Используется для возврата длины ASCII или двоичного файла. Эта функция часто используется совместно с POS и SEEK.
NAME	Name := File.NAME Используется для возврата имени ASCII или двоичного файла.
OPEN	[Ok] := File.OPEN Используется для открытия существующего ASCII или двоичного файла. По сравнению с CREATE, эта функция не создает файл, а открывает существующий. Необходимо всегда вызывать функцию TEXTMODE до использования CREATE или OPEN.
POS	Position := File.POS Используется для возврата текущей позиции указателя файла в ASCII или двоичном файле. Эта функция часто используется совместно с LEN и SEEK.

Microsoft Dynamics NAV 5.0 Quick Reference RU

QUERYREPLACE	<p>[IsQueryreplace] := File.QUERYREPLACE([SetQueryreplace]) Используется для того, чтобы определить, должна ли система спрашивать у пользователя разрешение на перезапись существующего файла. Также может использоваться для определения текущих настроек опций для файла.</p>
READ	<p>[Read] := File.READ(Variable) Используется для чтения из ASCII или двоичного файла. Если TEXTMODE установлен как TRUE, система прочтет строку текста из файла, оценит его и установит переменную, приравнивая ее к результату. Если TEXTMODE установлен как FALSE, система определит число байт для чтения, основываясь на размере переменной.</p>
RENAME	<p>[Ok] := File.RENAME(OldName, NewName) Используется для переименования ASCII или двоичного файла.</p>
SEEK	<p>File.SEEK(Position) Используется для установки указателя файла на новую позицию в ASCII или двоичном файле.</p>
SETSTAMP	<p>[Ok] := File.SETSTAMP(Name, Date [, Time]) Используется для установки метки времени для файла.</p>
TEXTMODE	<p>[IsTextmode] := File.TEXTMODE([SetTextmode]) Используется для определения, должен ли файл быть открыт как файл ASCII или двоичный файл. Также может использоваться для определения текущих настроек опций для файла.</p>
TRUNC	<p>File.TRUNC Используется для сокращения ASCII или двоичного файла до текущей позиции указателя файла.</p>
WRITE	<p>File.WRITE(Value) Используется для записи ASCII или двоичного файла. Если TEXTMODE установлен как TRUE и Value как integer, система конвертирует integer в тип text и запишет результат на новой строке. Если Value установлен как record, система разделит каждое поле табуляцией. Если TEXTMODE установлен как FALSE и Value как integer, система запишет integer как integer с длиной 4 байта.</p>
WRITEMODE	<p>[IsWritemode] := File.WRITEMODE([SetWritemode]) Используется до функции OPEN для установки или определения, можно ли вам записать в файл при более поздних вызовах функций.</p>

FORM

Переменные этого комплексного типа данных хранят формы. Формы содержат простые элементы управления. Элементы управления используются для показа информации пользователю или приема информации от пользователя.

ACTIVATE	[Ok] := Form.ACTIVATE Делает форму или элемент управления активным.
ACTIVE	IsActive := Form.ACTIVE Используется для установки значения формы – активно оно или нет.
CAPTION	[CurrCaption] := Form.CAPTION([NewCaption]) Используется для возврата текущего заголовка объекта как строки, и установки нового заголовка объекта.
CLOSE	Form.CLOSE Закрывает текущую форму.
EDITABLE	[IsEditable] := Form.EDITABLE([SetEditable]) Возвращает текущую настройку свойства Editable формы и изменяет ее. Функция может использоваться на формах и элементах управления.
FORM	Subform := Form.FORM Используется для доступа к форме, являющейся субформой текущей формы. Для запуска локальной функции для формы, являющейся субформой текущей формы, используйте: CurrForm.NameOfSubformControl.FORM.LocalFunctionToExecute;
FORM.RUN	FORM.RUN(Number [, Record] [, Field]) Используется для создания и запуска объекта формы.
FORM.RUNMODAL	[Action] := Form.RUNMODAL(Number [, Record] [, Field]) Используется для создания, запуска и закрытия объекта формы. Система запускает форму модально. Возвращаемый код показывает действие, которое выбрал пользователь. Возможные возвращаемые значения: OK, Cancel, LookupOK, LookupCancel, Yes, No, Close, Helpform, RunObject, RunSystem.
GETRECORD	Form.GETRECORD(Record) Используется для получения текущей записи, показанной на форме.
HEIGHT	[CurrHeight] := Form.HEIGHT([NewHeight]) Используется для возврата текущего значения свойства Height формы или элемента управления и установки этого свойства в новое значение. Единица измерения - 1/100 мм.
LOGHEIGHT	[CurrLogHeight] := Form.LOGHEIGHT([NewLogHeight]) Используется для возврата текущего значения свойства LogHeight формы и установки этого свойства в новое значение. Свойство используется для определения логической высоты формы. Логический размер формы может быть больше физического, который система показывает, когда форма открыта. Единица измерения - 1/100 мм.

Microsoft Dynamics NAV 5.0 Quick Reference RU

LOGWIDTH	<p>[CurrLogWidth] := Form.LOGWIDTH([NewLogWidth]) Используется для определения логической ширины формы. Логический размер формы может быть больше физического, который система показывает, когда форма открыта. Единица измерения - 1/100 мм.</p>
LOOKUPMODE	<p>[CurrLookupMode] := Form.LOOKUPMODE([NewLookupMode]) Используется для возврата текущего значения свойства LookupMode формы и установки этого свойства в новое значение. Если "NewLookupMode" установлен как "No", система автоматически отключит действия LookupOK и LookupCancel.</p>
MAXIMIZEDONOPEN	<p>[CurrMaximized] := Form.MAXIMIZEDONOPEN([NewMaximized]) Используется для возврата текущего значения свойства MaximizedOnOpen формы и установки этого свойства в новое значение. Используется для того, чтобы указать, должна ли система раскрыть форму на максимальный размер сразу после открытия.</p>
MINIMIZEDONOPEN	<p>[CurrMinimized] := Form.MINIMIZEDONOPEN([NewMinimized]) Используется для возврата текущего значения свойства MinimizedOnOpen формы и установки этого свойства в новое значение. Используется для того, чтобы указать, должна ли система установить форму в минимальный размер сразу после открытия.</p>
OBJECTID	<p>String := Form.OBJECTID([UseNames]) Возвращает строку в формате "form xxx", где xxx – это имя или номер объекта приложения.</p>
RUN	<p>Form.RUN Используется для запуска формы. Для очистки формы нужно использовать CLEAR.</p>
RUNMODAL	<p>[Action] := Form.RUNMODAL Используется для создания, запуска и закрытия формы. Возвращаемый код указывает принятое пользователем решение. Возможные возвращаемые значения: OK, Cancel, LookupOK, LookupCancel, Yes, No, Close, Helpform, RunObject, RunSystem.</p>
SAVERECORD	<p>CurrForm.SAVERECORD Используется для сохранения текущей записи, показанной на форме.</p>
SETRECORD	<p>Form.SETRECORD(Record) Используется для выбора текущей записи формы.</p>
SETSELECTIONFILTER	<p>CurrForm.SETSELECTIONFILTER(Record) Используется для получения системой записей, выбранных пользователем на форме, пометкой этих записей в таблицу и установки фильтра "только отмеченные". Если ни одна из записей не выбрана, отметки не будут использоваться.</p>
SETTABLEVIEW	<p>SETTABLEVIEW(Record) Используется для применения представления таблицы (Table View) на текущую запись как tableview для формы, отчета или dataport'a.</p>
UPDATE	<p>CurrForm.UPDATE([SaveRecord]) Используется для сохранения текущей записи и обновления элементов управления на форме. Если вы установите параметр SaveRecord как FALSE, эта функция не будет сохранять запись до тех пор, пока система не обновит форму.</p>

Microsoft Dynamics NAV 5.0 Quick Reference RU

UPDATECONTROLS	<code>CurrForm.UPDATECONTROLS</code> Используется для перезагрузки заголовков всех элементов управления на форме. Это необходимо, когда пользователь изменяет класс заголовков после загрузки формы.
UPDATEEDITABLE	<code>UPDATEEDITABLE(Editable)</code> Используется для динамического изменения свойства <code>Editable</code> (редактируемый) поля, формы или элемента управления. Может вызываться только из <code>OnBeforeInput</code> элемента управления.
URL	<code>String :=Form.URL([UseNames])</code> Возвращает строку, содержащую полную ссылку (URL) к форме.
VISIBLE	<code>[IsVisible] := Form.VISIBLE([SetVisible])</code> Используется для возврата текущего свойства <code>Visible</code> (видимый) формы или элемента управления и изменения этого свойства.
WIDTH	<code>[CurrWidth] := Form.WIDTH([NewWidth])</code> Используется для возврата текущего свойства <code>Width</code> (ширина) формы или элемента управления и установки этого свойства в новое значение. Единица измерения - 1/100 мм.
XPOS	<code>[CurrXPos] := Form.XPOS([NewXPos])</code> Используется для возврата текущего свойства <code>XPos</code> (горизонтальная позиция) формы или элемента управления и установки этого свойства в новое значение. Единица измерения - 1/100 мм.
YPOS	<code>[CurrYPos] := Form.YPOS([NewYPos])</code> Используется для возврата текущего свойства <code>YPos</code> (вертикальная позиция) формы или элемента управления и установки этого свойства в новое значение. Единица измерения - 1/100 мм.

GUID

Используйте данный тип данных, чтобы задать уникальный идентификационный номер любому объекту БД. Тип данных *Globally Unique Identifier (GUID, Уникальный Глобальный Идентификатор)* имеет длину 16 байт (двоичный тип данных). Он используется для глобальной идентификации объектов, программ, записей и т.п. Важным свойством *GUID* является то, что каждое значение уникально. Значение генерируется алгоритмом, разработанным Microsoft, гарантирующим уникальность. Стандартное текстовое представление - {12345678-1234-1234-1234-1234567890A}

CREATEGUID	<code>Guid :=CREATEGUID()</code> Создает новый уникальный GUID. Его значение затем можно назначить как тип данных GUID или текстовый тип данных. Используйте текстовый тип данных, если хотите сравнить GUID с другой текстовой строкой.
ISNULLGUID	<code>Ok := ISNULLGUID(Guid)</code> Используется для проверки, было ли назначено значение для GUID.Null (пустой) GUID, содержащий только нули – верный, но он никогда не должен использоваться при работе на реальных объектах.

INSTREAM & OUTSTREAM

Типы данных *InStream* (input stream, входящий поток) и *OutStream* (output stream, исходящий поток) – это потоковые объекты, которые можно использовать для чтения или записи в файлы и BLOBы. Кроме этого, *InStream* и *OutStream* активируют данные для чтения и посылают их на объекты типа *Automation* и *OCX*. *Microsoft XML DOM* может выполнять чтение из объекта *InStream* и записывать на объект *OutStream*.

InStream.EOS	IsEOS := InStream.EOS() Используется для получения информации о том, достиг ли конца потока (End of Stream, EOS) входящий поток.
InStream.READ	[{Read}] := InStream.Read(Variable, [Length]) Используется для чтения специального набора байт из объекта <i>InStream</i> . Чтение данных выполняется в двоичном формате.
InStream.READTEXT	[{Read}] := InStream.ReadText(Text, [Length]) Используется для чтения текста из объекта <i>InStream</i> . READTEXT выполняет чтение специального набора байт, максимальной длины строки или пока не достигнет конца строки. Чтение данных выполняется в текстовом формате.
OutStream.WRITE	[{Written}] := OutStream.Write(Variable, [Length]) Используется для записи специального набора в поток. Запись данных выполняется в двоичном формате.
OutStream.WRITETEXT	[{Written}] := OutStream.WriteText([Text, [Length]]) Используется для записи текста в объект <i>OutStream</i> . Запись данных выполняется в текстовом формате.

KEYREF

Этот комплексный тип данных идентифицирует ключ в таблице и поля в ключе. Это дает доступ к ключу и содержащим его полям. Объект *keyref* (индекс ключа) может ссылаться на любой ключ любой таблицы БД.

ACTIVE	Ok := KeyRef.ACTIVE Позволяет узнать, включен ключ или нет.
FIELD COUNT	No := KeyRef.FIELD COUNT Используется для возврата числа полей, определенных в ключе. Функция возвращают ошибку в случае, если не выбран ключ.
FIELD INDEX	FieldRef := KeyRef.FIELD INDEX(Index) Используется для возврата <i>fieldref</i> (индекс) поля.
RECORD	RecordRef := KeyRef.RECORD Используется для возврата индекса записи (<i>recordref</i>) ключа. Функция возвращают ошибку в случае, не выбран ключ.

NUMERIC

Существуют следующие типы числовых данных: *BIGINTEGER, DECIMAL, INTEGER*.

ABS	<p><code>NewNumber := ABS(Number)</code> Используется для вычисления абсолютного значения (модуля) числа. ABS всегда возвращает положительное значение или ноль.</p>
POWER	<p><code>NewNumber := POWER(Number, Power)</code> Используется для возведения числа (Number) в степень (Power).</p>
RANDOM	<p><code>Number := RANDOM(MaxNumber)</code> Используется для возврата псевдослучайного числа.</p>
RANDOMIZE	<p><code>RANDOMIZE([Seed])</code> Используется для генерации набора случайных чисел, из которого RANDOM выберет случайный номер. "Seed" (отбор) – это число, которое RANDOMIZE использует для создания уникального набора чисел. Если вы используете seed повторно, вы получите тот же самый набор случайных чисел. Если не заполнять опциональный параметр, RANDOMIZE будет использовать текущее системное время (общее число миллисекунд, начиная с полуночи).</p>
ROUND	<p><code>NewNumber := ROUND(Number [, Precision] [, Direction])</code> Используется для округления значения числовой переменной. Параметр Precision (точность) определяет, с какой точностью округлять. Значение по умолчанию - 0.01 для английской версии. В других странах/регионах могут использоваться другие значения. Параметр "Direction" сообщает системе, как округлять число (Number). Метод округления по умолчанию - '='. Всего же существуют 3 метода округления: '=', '>', '<'</p>

RECORD

Этот комплексный тип данных записывает данные в строку таблицы. Каждая запись состоит из полей (формирующих столбцы таблицы). Запись обычно используется для хранения информации о фиксированном числе свойств.

ADDLINK [5.0]	<p><code>[ID] := ADDLINK(URL[, Description])</code> Используется для добавления ссылки к записи. Каждый раз, когда вы добавляете ссылку к форме или таблице, в системной таблице Record Link создается запись. URL может быть ссылкой на веб-сайт, на файл, хранящийся на локальном или удаленном компьютере, или ссылкой на форму Dynamics NAV.</p>
ASCENDING	<p><code>[IsAscending] := Record.ASCENDING([SetAscending])</code> Используется для изменения или проверки порядка, в котором система будет выполнять поиск через таблицу C/SIDE.</p>
CALCFIELDS	<p><code>[Ok] := Record.CALCFIELDS(Field1, [Field2],...)</code> Используется для вычисления FlowFields в записи.</p>
CALCSUMS	<p><code>[Ok] := Record.CALCSUMS(Field1, [Field2],...)</code> Используется для вычисления общего числа столбцов SumIndexFields в таблице C/SIDE.</p>

Microsoft Dynamics NAV 5.0 Quick Reference RU

CHANGECOMPANY	<p>[Ok] := Record.CHANGECOMPANY([CompanyName])</p> <p>Используется для перенаправления ссылки к данным таблицы из одной фирмы в другую. Функция CHANGECOMPANY не затрагивается функцией RESET. Вы можете отменить выбор фирмы, снова вызвав CHANGECOMPANY или используя CLEAR.</p>
CLEARMARKS	<p>Record.CLEARMARKS</p> <p>Используется для удаления всех меток на записи.</p>
CONSISTENT	<p>Record.CONSISTENT(Consistent)</p> <p>Используется для пометки таблицы C/SIDE как совместимой или несовместимой с административной точки зрения. Обычно эта функция используется только в бухгалтерской работе.</p>
COPY	<p>Record.COPY(FromRecord)</p> <p>Используется для копирования записи из таблицы. Все фильтры, метки и ключи включаются в копию.</p>
COPYFILTER	<p>Record.COPYFILTER(FromField, ToRecord.ToField)</p> <p>Используется для копирования фильтра одного поля и применения его к другому полю.</p>
COPYFILTERS	<p>Record.COPYFILTERS(FromRecord)</p> <p>Используется для копирования всех фильтров от SETFILTER или SETRANGE из одной записи в другую.</p>
COPYLINKS [5.0]	<p>COPYLINKS(FromRecord)</p> <p>Используется для копирования всех ссылок из записи.</p>
COUNT	<p>Number := Record.COUNT</p> <p>Используется для подсчета количества записей в таблице C/SIDE.</p>
COUNTAPPROX	<p>Number := Record.COUNTAPPROX</p> <p>Используется для получения приближенного количества чисел записи таблицы, например для обновления прогресс-бара или показа информационных сообщений. Количество приближено из-за того, что она использует статистическую информацию, поддерживаемой сервером SQL и не хранится в точном значении.</p>
CURRENTKEY	<p>CurrentKey := Record.CURRENTKEY</p> <p>Используется для возврата текущего ключа таблицы БД.</p>
DELETE	<p>[Ok] := Record.DELETE([RunTrigger])</p> <p>Используется для удаления записи в таблице C/SIDE. Булевый параметр "RunTrigger" позволяет запустить код C/AL в триггере OnDelete.</p>
DELETEALL	<p>Record.DELETEALL([RunTrigger])</p> <p>Используется для удаления всех записей в таблице C/SIDE со специальным диапазоном. Система удаляет только отфильтрованные записи. Булевый параметр "RunTrigger" позволяет запустить код C/AL в триггере OnDelete. Когда RunTrigger выставлен как FALSE (по умолчанию), функция DELETEALL работает намного быстрее, потому что вызывается только один раз. Если RunTrigger выставлен как TRUE, производительность системы будет невысокой, поскольку C/SIDE должен будет загружать каждую запись в любом случае, чтобы выполнить триггер OnDelete.</p>

Microsoft Dynamics NAV 5.0 Quick Reference RU

DELETEDLINK [5.0]	Record.DELETEDLINK(ID) Используется для удаления ссылки, добавленной к записи в таблице.
DELETEDLINKS [5.0]	Record.DELETEDLINKS Используется для удаления всех ссылок, добавленных к записи.
FIELDACTIVE	Ok := Record.FIELDACTIVE(Field) Используется для проверки, активно поле или нет.
FIELDCAPTION	Caption := Record.FIELDCAPTION(Field) Используется для возврата текущего заголовка поля в виде строки. FIELDCAPTION возвращает caption (заголовок) поля. FIELDCAPTION в первую очередь смотрит на свойство CaptionML. При его отсутствии будет использоваться свойство Name.
FIELDERROR	Record.FIELDERROR(Field, [Text]) Используется для прекращения выполнения кода и создания сообщения об ошибке для поля.
FIELDNAME	Name := Record.FIELDNAME(Field) Используется для возврата имени поля в виде строки.
FIELDNO	Number := Record.FIELDNO(Field) Используется для возврата номера, присвоенного полю в описании таблицы.
FILTERGROUP	[CurrGroup] := Record.FILTERGROUP([NewGroup]) Используется для выбора группы фильтров и нахождения номера текущей группы фильтров. Filtergroup может содержать фильтр для Record, установленный ранее с помощью SETFILTER или SETRANGE. Число примененных фильтров – это комбинация всех фильтров, установленных во всех группах фильтров. C/SIDE использует 7 внутренних групп фильтров: 0 Std (стандартный), 1 Global (глобальный), 2 Form (форма), 3 Eхес (исполняющий), 4 Link (ссылка), 5 Temp (временный), 6 Security (безопасность).
FIND	Ok := Record.FIND([Which]) Используется для нахождения записи в таблице C/SIDE, основанной на значениях, хранимых в ключах. "Which" сообщает системе, как производить поиск. Если SearchStr содержит '=', '>' или '<', вы должны назначить значения на все поля текущих и первичных ключей до того, как вызовете функцию FIND.
FINDFIRST	Ok := Record.FINDFIRST Используется для нахождения первой записи в таблице, основанной на текущем ключе и фильтре. Эта функция должна использоваться вместо FIND('-'), когда вам нужна только первая запись. Не используйте функцию совместно с REPEAT..UNTIL.
FINDLAST	Ok := Record.FINDLAST Используется для нахождения последней записи в таблице, основанной на текущем ключе и фильтре. Эта функция должна использоваться вместо FIND('+'), когда вам нужна только последняя запись. Не используйте функцию совместно с REPEAT..UNTIL.

Microsoft Dynamics NAV 5.0 Quick Reference RU

FINDSET	<p>Ok := Record.FINDSET([ForUpdate][, UpdateKey])</p> <p>Используется для нахождения набора записей в таблице, основанной на текущем ключе и фильтре. Записи можно получить только в порядке возрастания. Установите "ForUpdate" как FALSE, если не намереваетесь модифицировать записи в наборе. Установите "ForUpdateKey" как TRUE, если собираетесь модифицировать любое значение поля совместно с ключом. Используйте эту функцию только если хотите явно пройти цикл через набор записей. Вы должны использовать эту функцию ТОЛЬКО совместно с REPEAT .. UNTIL</p>
GET	<p>[Ok] := Record.GET([Value] ,...)</p> <p>Используется для нахождения записи, основанной на значениях, хранимых в первичных ключевых полях. Эта функция всегда использует первичный ключ для таблицы и игнорирует любые фильтры. После вызова система не изменяет текущий ключ и фильтры.</p>
GETFILTER	<p>String := Record.GETFILTER(Field)</p> <p>Используется для возврата списка фильтров совместно с текущей группой фильтра, примененной к полю.</p>
GETFILTERS	<p>String := Record.GETFILTERS</p> <p>Используется для возврата строки, содержащей список фильтров совместно с текущей группой фильтров для всех полей в записи. Плюс ко всему, эта функция также возвращает состояние MARKEDONLY.</p>
GETPOSITION	<p>String := Record.GETPOSITION([UseNames])</p> <p>Используется для возврата строки, содержащей первичный ключ текущей записи.</p>
GETRANGEMAX	<p>Value := Record.GETRANGEMAX(Field)</p> <p>Используется для возврата максимального значения диапазона для поля. Тип данных значения должен соответствовать типу поля.</p>
GETRANGEMIN	<p>Value := Record.GETRANGEMIN(Field)</p> <p>Используется для возврата минимального значения диапазона для поля. Тип данных значения должен соответствовать типу поля.</p>
GETVIEW	<p>String := Record.GETVIEW([UseNames])</p> <p>Используется для возврата строки, которая описывает текущий метод сортировки, ключ и фильтры на таблице.</p>
HASFILTER	<p>Ok := Record.HASFILTER</p> <p>Используется для определения, назначила ли система фильтр на запись совместно с текущей группой фильтра.</p>
HASLINKS [5.0]	<p>Ok := Record.HASLINKS</p> <p>Используется для определения, содержит ли запись какие-нибудь ссылки.</p>
INIT	<p>Record.INIT</p> <p>Используется для инициализации записи в таблице C/SIDE. Система не инициализирует первичный ключ или поля с временной меткой.</p>

Microsoft Dynamics NAV 5.0 Quick Reference RU

ISEMPTY	<p>Empty := Record.ISEMPTY</p> <p>Используется для определения, пуста ли таблица C/SIDE или отфильтрованный набор записей. При использовании SQL Server, эта функция работает быстрее, чем Record.COUNT и затем тестирует результат на наличие нулей.</p>
INSERT	<p>[Ok] := Record.INSERT([RunTrigger])</p> <p>Используется для вставки записи в таблицу C/SIDE. Если параметр "RunTrigger" установлен как TRUE, то будет выполняться код в триггере OnInsert.</p>
LOCKTABLE	<p>Record.LOCKTABLE([Wait] [, VersionCheck])</p> <p>Используется для блокирования таблицы C/SIDE с целью защиты от транзакций записи, конфликтующих друг с другом. SQL Server Option для Microsoft Dynamics NAV поддерживает только значения по умолчанию для параметров функции LOCKTABLE – LOCKTABLE(TRUE,FALSE).</p>
MARK	<p>[IsMarked] := Record.MARK([SetMarked])</p> <p>Используется для метки записи. Можно также использовать функцию, чтобы узнать, имеет ли запись метку.</p>
MARKEDONLY	<p>[IsMarkedOnly] := Record.MARKEDONLY([SetMarkedOnly])</p> <p>Сообщает системе о необходимости активации специального фильтра. После использования данной функции, представление таблицы будет включать только записи, помеченные этой функцией.</p>
MODIFY	<p>[Ok] := Record.MODIFY([RunTrigger])</p> <p>Используется для модификации записей в таблице C/SIDE. Если параметр "RunTrigger" установлен как TRUE, будет выполняться код в триггере OnModify.</p>
MODIFYALL	<p>Record.MODIFYALL(Field, NewValue [, RunTrigger])</p> <p>Используется для модификации полей во всех записях в заданном диапазоне. Если параметр "RunTrigger" установлен как TRUE, будет выполняться код в триггере OnModify.</p>
NEXT	<p>Steps := Record.NEXT([Steps])</p> <p>Используется для прохождения через заданное число записей до нужной. Steps (шаги) используются для определения направления поиска и количества записей для перешагивания. > 0: поиск вверх по таблице. < 0: поиск вниз по таблице. = 0: Без определения направления. В этом случае система будет находить следующую ближайшую запись.</p>
READCONSISTENCY	<p>Ok := Record.READCONSISTENCY</p> <p>Используется для определения, поддерживает ли таблица последовательность чтения. Сервер БД Microsoft Dynamics NAV поддерживает последовательность чтения как неотъемлемую часть архитектуры БД. Сервер SQL не поддерживает данную особенность.</p>
READPERMISSION	<p>Ok := Record.READPERMISSION</p> <p>Используется для установки разрешения (прав) на чтение из таблицы. Может протестировать на полное и частичное разрешения на чтение, устанавливаемое security filter (фильтром безопасности).</p>

Microsoft Dynamics NAV 5.0 Quick Reference RU

RECORDLEVELLOCKING	<p>Ok := Record.RECORDLEVELLOCKING</p> <p>Используется для определения, поддерживает ли таблица блокировку уровня записи (record level locking). При использовании SQL Server, вы можете использовать record level locking. При использовании Microsoft Dynamics NAV Database Server, вы не можете использовать record level locking.</p>
RELATION	<p>TableNumber := Record.RELATION(Field)</p> <p>Используется для нахождения связи таблиц заданного поля.</p>
RENAME	<p>[Ok] := Record.RENAME(Value1, [Value2],...)</p> <p>Используется для изменения первичного ключа в таблице C/SIDE.</p>
RESET	<p>Record.RESET</p> <p>Используется для удаления всех фильтров, включая любые специальные фильтры, установленные с помощью MARKEDONLY, и установки текущего ключа как первичного. Система также удаляет любые метки на записи и чистит любые переменные C/AL на записи.</p>
SETCURRENTKEY	<p>[Ok] := Record.SETCURRENTKEY(Field1, [Field2],...)</p> <p>Используется для выбора ключа таблицы. Во время поиска ключа, C/SIDE выбирает первое событие определенных полей.</p>
SETFILTER	<p>Record.SETFILTER(Field, String,[Value],...)</p> <p>Используется для наложения фильтра на поле.</p>
SETPERMISSIONFILTER	<p>Record.SETPERMISSIONFILTER</p> <p>Используется для применения фильтра пользовательской безопасности на переменную Record. Фильтр безопасности может работать совместно с любыми другими фильтрами, размещенными на переменной Record с помощью SETFILTER или SETRANGE. Эта функция C/AL применяется только при работе в режиме SQL Server Option для Microsoft Dynamics NAV.</p>
SETPOSITION	<p>Record.SETPOSITION(String)</p> <p>Используется для установки поля в первичном ключе на записи. Оставшиеся поля остаются нетронутыми.</p>
SETRANGE	<p>Record.SETRANGE(Field [,FromValue] [,ToValue])</p> <p>Используется для установки на поле простого фильтра, такого как простой диапазон или простое значение. Если использовать SETRANGE без настройки параметров From-Value/To-Value, функция удалит любые уже установленные фильтры.</p>
SETRECFILTER	<p>Record.SETRECFILTER</p> <p>Используется для установки значений в текущем ключе текущей записи как фильтр записи. Можно использовать функцию для установки фильтра на таблицу до запуска отчета.</p>
SETVIEW	<p>Record.SETVIEW(String)</p> <p>Используется для установки текущего метода сортировки, ключа и фильтров на таблице. Если функция SETVIEW выполняется с пустой строкой (параметром String), то все фильтры удалятся и будет использоваться первичный ключ.</p>

Microsoft Dynamics NAV 5.0 Quick Reference RU

TABLECAPTION	Caption := Record.TABLECAPTION Используется для возврата текущего заголовка таблицы в виде строки. TABLECAPTION в первую очередь смотрит на свойство CaptionML. Если там будет пусто, то будет использоваться свойство Name.
TABLENAME	Name := Record.TABLENAME Используется для возврата имени таблицы C/SIDE.
TESTFIELD	Record.TESTFIELD(Field, [Value]) Используется, чтобы посмотреть, соответствует ли содержимое поля заданному значению. Если содержимое отличается от заданного значения, система выведет сообщение об ошибке. Если параметр Value и Field указать как 0 или оставить пустым (пустая строка), система также выведет сообщение об ошибке.
TRANSFERFIELDS	Record.TRANSFERFIELDS(FromRecord [, InitPrimaryKeyFields]) Используется для копирования всех соответствующих полей одной записи на другую. Поля копируются, основываясь на свойстве Field No. Если вы копируете запись, содержащую поле BLOB, вы должны вычислить поле BLOB до того, как оно может быть скопировано записью.
VALIDATE	Record.VALIDATE(Field [, NewValue]) Используется для вызова триггеров для поля.
WRITEPERMISSION	Ok := Record.WRITEPERMISSION Используется для определения прав записи в таблице. Может протестировать на полное и частичное разрешения на чтение, устанавливаемое security filter (фильтром безопасности). Права записи состоят из таких прав, как Insert, Delete и Modify.

RECORDID

Этот тип данных содержит номер таблицы и первичный ключ таблицы. Вы можете хранить RecordID в БД, но не можете устанавливать фильтры на RecordID.

GETRECORD	RecordRef := RecordID.GETRECORD Используется для возврата recordref (индекса записи).
TABLENO	No := RecordID.TABLENO Используется для возврата номера таблицы, определенного с помощью recordid. Функция возвращает ошибку, если запись пуста.

RECORDREF

Этот комплексный тип данных идентифицирует строку в таблице. Каждая запись состоит из полей (формирующих столбцы в таблице). Запись обычно используется для содержания информации о фиксированном количестве свойств. Объект RecordRef может ссылаться на любую таблицу в БД. Используйте функцию RecordRef.OPEN, чтобы выбрать таблицу, к которой вы хотите получить доступ. При использовании RecordRef.OPEN создается новый объект. Этот объект содержит ссылки для открытия таблиц, фильтров и записей и всех содержащихся в нем полей.

ADDLINK [5.0]	[ID] := ADDLINK(URL[, Description]) Используется для добавления ссылки на запись в таблице. URL может быть ссылкой на веб-сайт, файл, хранящийся на локальном или удаленном компьютере или ссылкой на форму Navision.
ASCENDING	[IsAscending] := RecordRef.ASCENDING([SetAscending]) Используется для изменения или установки способа, в котором система будет производить поиск через таблицу.
CAPTION	Caption := RecordRef.CAPTION Используется для возврата заголовка текущей выбранной таблицы. Если не выбрана ни одна таблица, то функция возвратит ошибку.
CLOSE	RecordRef.CLOSE Используется для закрытия текущей таблицы.
COUNT	Number := RecordRef.COUNT Используется для подсчета числа записей.
COPYLINKS [5.0]	COPYLINKS(FromRecord) Используется для копирования всех ссылок из уникальной записи.
COUNTAPPROX	Number := RecordRef.COUNTAPPROX Используется для получения приближенного числа записей в таблице, например, для обновления прогресс-бара или показа информационных сообщений. Число приближенно из-за использования статистической информации, поддерживаемой SQL Server, и не хранимой в точном виде.
CURRENTKEY	CurrentKey := RecordRef.CURRENTKEY Используется для возврата текущего ключа таблицы. Используемый тип данных при возврате - строка.
CURRENTKEYINDEX	[CurrKeyIndex] := RecordRef.CURRENTKEYINDEX([NewKeyIndex]) Используется для возврата или установки индекса ключа таблицы. Индекс ключа устанавливается или возвращается в числовом виде.
DELETE	[Ok] := RecordRef.DELETE([RunTrigger]) Используется для удаления записи в таблице C/SIDE. Опциональный параметр позволяет запускать код C/AL в триггере OnDelete.

Microsoft Dynamics NAV 5.0 Quick Reference RU

DELETEALL	<p>RecordRef.DELETEALL([RunTrigger])</p> <p>Используется для удаления всех записей в таблице C/SIDE в заданном диапазоне. Система удаляет только те записи, которые отобраны по фильтру для recordref. Опциональный параметр позволяет запускать код C/AL в триггере OnDelete.</p>
DELETEDLINK [5.0]	<p>DELETEDLINK(ID)</p> <p>Используется для удаления ссылки, прикрепленной к записи в таблице.</p>
DELETEDLINKS [5.0]	<p>DELETEDLINKS</p> <p>Используется для удаления всех ссылок, прикрепленных к записи.</p>
DUPLICATE	<p>RecordRef := RecordRef.DUPLICATE</p> <p>Используется для дубликации таблицы, содержащей recordref.</p>
FIELD	<p>Field := RecordRef.FIELD(FieldNo)</p> <p>Используется для возврата recordref поля, имеющего номер fieldno в текущей выбранной таблице. Если поля с таким номером нет, функция возвратит ошибку.</p>
FIELDCOUNT	<p>Count := RecordRef.FIELDCOUNT</p> <p>Используется для возврата числа полей в текущей выбранной таблице или для возврата числа полей, определенных в ключе. В случае, если не выбрана таблица или ключ, эти функции возвращают ошибку.</p>
FIELDEXIST	<p>Exist := RecordRef.FIELDEXIST(FieldNo)</p> <p>Используется для определения существования поля с номером fieldno. В случае, если не выбрана таблица, эта функция возвращает ошибку.</p>
FIELDINDEX	<p>Field := RecordRef.FIELDINDEX(Index)</p> <p>Используется для возврата fieldref (индекса) поля.</p>
FILTERGROUP	<p>[CurrGroup] := RecordRef.FILTERGROUP([NewGroup])</p> <p>Используется для изменения группы фильтров, примененных к таблице. Filtergroup может содержать фильтр для RecordRef, установленного ранее с помощью SETFILTER или SETRANGE. Количество примененных фильтров – это комбинация всех фильтров, установленных во всех группах фильтра.</p>
FIND	<p>[Ok] := RecordRef.FIND([Which])</p> <p>Используется для нахождения записи в таблице на основании значений, хранимых в ключевых полях.</p>
FINDFIRST	<p>Ok := Record.FINDFIRST</p> <p>Используется для нахождения первой записи в таблице. Вы должны использовать эту функцию только тогда, когда хотите в явном виде найти первую запись в таблице. Не используйте функцию совместно с REPEAT .. UNTIL.</p>
FINDLAST	<p>Ok := Record.FINDLAST</p> <p>Используется для нахождения последней записи в таблице. Вы должны использовать эту функцию только тогда, когда хотите в явном виде найти последнюю запись в таблице. Не используйте функцию совместно с REPEAT .. UNTIL.</p>

Microsoft Dynamics NAV 5.0 Quick Reference RU

FINDSET	<p>Ok := Record.FINDSET([ForUpdate][, UpdateKey]) Используется для нахождения набора записей в таблице на основании текущего ключа и фильтра. Записи можно искать только в порядке возрастания.</p>
GET	<p>[Ok] :=RecordRef.GET(RecordID) Используется для нахождения записи на основании ID.</p>
GETFILTERS	<p>String := RecordRef.GETFILTERS Используется для определения, какие фильтры были применены к таблице.</p>
GETPOSITION	<p>String := RecordRef.GETPOSITION([UseNames]) Используется для возврата строки, содержащей первичный ключ текущей записи.</p>
GETTABLE	<p>RecordRef.GETTABLE(rec) Присваивает переменной RecordRef ссылку на таблицу, с которой будут производиться какие-либо действия.</p>
GETVIEW	<p>String := RecordRef.GETVIEW([UseNames]) Используется для возврата строки, описывающей текущий порядок сортировки, ключ и фильтры к таблице.</p>
HASFILTER	<p>Ok := RecordRef.HASFILTER Используется для определения, был ли применен фильтр к таблице.</p>
HASLINKS [5.0]	<p>Ok := HASLINKS Используется для определения, содержит ли запись какие-либо ссылки.</p>
INIT	<p>RecordRef.INIT Используется для инициализации записи в таблице. Система не инициализирует первичный ключ или поля с временной меткой.</p>
INSERT	<p>[Ok] := RecordRef.INSERT([RunTrigger]) Используется для вставки записи в таблицу. Булевый параметр "RunTrigger" позволяет запустить код C/AL на триггере OnInsert.</p>
ISEMPTY	<p>Empty := RecordRef.ISEMPTY Используется для определения, существуют ли какие-либо записи совместно с фильтрованным набором записей в таблице.</p>
KEYCOUNT	<p>Count := RecordRef.KEYCOUNT Используется для возврата числа ключей, существующих в таблице. Если таблица не выбрана, функция возвратит ошибку.</p>
KEYINDEX	<p>Key := RecordRef.KEYINDEX(Index) Используется для возврата индекса ключа текущей выбранной таблицы.</p>
LOCKTABLE	<p>RecordRef.LOCKTABLE([Wait] [, VersionCheck]) Используется для блокирования таблицы в целях защиты от транзакций записи, конфликтующих друг с другом.</p>
MODIFY	<p>[Ok] := RecordRef.MODIFY([RunTrigger]) Используется для модификации записи в таблице C/SIDE.</p>

Microsoft Dynamics NAV 5.0 Quick Reference RU

NAME	Name := RecordRef.NAME Используется для возврата имени текущей выбранной таблицы. Если таблица не выбрана, функция возвратит ошибку.
NEXT	[Steps] := RecordRef.NEXT([Steps]) Используется для перешагивания определенного числа записей и получения нужной.
NUMBER	No := RecordRef.NUMBER Используется для возврата ID таблицы (номера), содержащей запись.
OPEN	RecordRef.OPEN(No[, Temp][, CompanyName]) Используется для создания и открытия переменной RecordRef, идентифицируемой номером в фирме. Если опустить этот параметр, система будет использовать текущую фирму.
READCONSISTENCY	Ok := RecordRef.READCONSISTENCY Используется для того, чтобы узнать, поддерживается ли последовательность чтения. Сервер БД Microsoft Dynamics NAV поддерживает последовательность чтения как неотъемлемую часть архитектуры БД, тогда как сервер SQL эту особенность не поддерживает.
READPERMISSION	Ok := RecordRef.READPERMISSION Используется для определения разрешения (прав) на чтение из таблицы. Может протестировать на полное и частичное разрешения на чтение, устанавливаемые security filter (фильтром безопасности).
RECORDID	RecordID := RecordRef.RECORDID Используется для возврата RecordID текущей выбранной записи таблицы. Если таблица не выбрана, то сгенерируется ошибка.
RECORDLEVELLOCKING	Ok := RecordRef.RECORDLEVELLOCKING Используется для определения поддержки таблицей блокировки уровня записей. При использовании сервера SQL блокировка уровня записей поддерживается, а при использовании сервера БД такой возможностью воспользоваться нельзя.
RESET	RecordRef.RESET Используется для удаления всех фильтров, включая любые специальные фильтры, установленные с помощью MARKEDONLY и изменения статуса текущего ключа на первичный. Система также удаляет любые метки на записи и очищает любые переменные C/AL на записи.
SETPERMISSIONFILTER	RecordRef.SETPERMISSIONFILTER Используется для применения пользовательского фильтра безопасности на переменную RecordRef. Фильтр безопасности сочетается с любыми другими фильтрами, наложенными на переменную RecordRef с помощью SETFILTER или SETRANGE.
SETPOSITION	RecordRef.SETPOSITION(String) Используется для установки полей в первичном ключе в записи на значения, определенные в строке. Другие поля остаются нетронутыми.

Microsoft Dynamics NAV 5.0 Quick Reference RU

SETRECFILTER	RecordRef.SETRECFILTER Используется для установки фильтра на запись.
SETTABLE	RecordRef.SETTABLE(rec) Используется для создания переменной записи в той же таблице, в которой находится индекс записи.
SETVIEW	RecordRef.SETVIEW(String) Используется для установки текущего порядка сортировки, ключа и фильтров на таблице.
WRITEPERMISSION	Ok := RecordRef.WRITEPERMISSION Используется для определения разрешения (прав) на запись в таблицу. Может протестировать на полное и частичное разрешения на запись, устанавливаемые фильтром безопасности. Права записи состоят из прав Insert (вставки), Delete (удаления) and Modify (изменения).

REPORT

Используйте этот комплексный тип данных для хранения отчетов (reports). Отчеты состоят из простых элементов, называемых элементами управления. Элементы управления используются для вывода информации пользователю или приема информации от пользователя

BREAK	BREAK Используется для выхода из цикла или триггера dataport'а, отчета или XMLport'а.
CREATETOTALS	CREATETOTALS(Var1 [, Var2] ,...) Используется для создания переменных.
NEWPAGE	NEWPAGE Используется для создания новой страницы при печати отчета.
NEWPAGEPERRECORD	[IsNewPagePerRecord] := NEWPAGEPERRECORD([SetNewPagePerRecord]) Используется для возврата текущего свойства NewPagePerRecord (новая страница на каждую запись) и установки этого свойства в новое значение.
OBJECTID	String :=Report.OBJECTID([UseNames]) Используется для возврата имени отчета. Если параметр "UseNames" установлен как TRUE (значение по умолчанию) или вообще пуст, возвращаемая строка будет содержать имя отчета. В противном случае (FALSE) в строке будет содержаться номер отчета.
PAGENO	[CurrPageNo] := PAGENO([NewPageNo]) Используется для возврата текущего номера страницы у отчета и установки нового номера страницы.
PAPERSOURCE	CurrReport.PAPERSOURCE(PaperBinNo [, PhysicalPage]) Используется для возврата источника бумаги, используемого для текущей или другой страницы и установки нового источника бумаги.

Microsoft Dynamics NAV 5.0 Quick Reference RU

PREVIEW	<p>IsPreview := PREVIEW</p> <p>Используется для определения, будет ли формироваться отчет в режиме предварительного просмотра.</p>
PRINTONLYIFDETAIL	<p>[IsPrintOnlyIfDetail] := PRINTONLYIFDETAIL([SetPrintOnlyIfDetail])</p> <p>Используется для возврата текущего свойства PrintOnlyIfDetail (печатать только если есть данные) и установки этого свойства в новое значение.</p>
QUIT	<p>QUIT</p> <p>Используется для прерывания выполнения dataport'а, отчета или XMLport'а. При использовании функции QUIT dataport, отчет или XMLport остаются без каких-либо изменений, которые были сделаны во время их выполнения в БД. Триггеры OnPostReport, OnPostDataPort или OnPostXMLport не вызываются.</p>
REPORT.RUN	<p>REPORT.RUN(Number [, ReqWindow] [, SystemPrinter] [, Record])</p> <p>Используется для загрузки и выполнения отчета. Параметр "SystemPrinter" сообщает системе, какой она должна использовать принтер - принтер Windows по умолчанию или принтер из таблицы Printer Selection.</p>
REPORT.RUNMODAL	<p>REPORT.RUNMODAL(Number [, ReqWindow] [, SystemPrinter] [, Record])</p> <p>Используется для загрузки и выполнения отчета. Параметр "SystemPrinter" сообщает системе, какой она должна использовать принтер - принтер Windows по умолчанию или принтер из таблицы Printer Selection.</p>
RUN	<p>Report.RUN</p> <p>Используется для загрузки и выполнения отчета. После выполнения этой функции система автоматически чистит переменные.</p>
RUNMODAL	<p>Report.RUNMODAL</p> <p>Используется для загрузки и выполнения отчета. После выполнения этой функции система не выполняет автоматическую чистку переменных.</p>
SAVEASHTML	<p>[Ok] := Report.SAVEASHTML(Number, FileName [,SystemPrinter] [, Rec])</p> <p>[Ok] := Report.SAVEASHTML(FileName)</p> <p>Используется для сохранения отчета в виде файла HTML. Для просмотра файла рекомендуется использовать браузер с поддержкой HTML версии 3.0 или более поздней.</p>
SAVEASXML	<p>[Ok] := Report.SAVEASXML(Number, FileName [,SystemPrinter] [, Rec])</p> <p>[Ok] := Report.SAVEASXML(FileName)</p> <p>Используется для сохранения отчета в виде файла XML.</p>
SETTABLEVIEW	<p>SETTABLEVIEW(Record)</p> <p>Используется для применения Table View (представления таблицы) на текущую запись.</p>
SHOWOUTPUT	<p>[IsShow] := SHOWOUTPUT ([SetShow])</p> <p>Используется для возврата значения настройки вывода секции и изменения этой настройки.</p>
SKIP	<p>SKIP</p> <p>Используется для пропуска текущей итерации dataport'а, отчета или XMLport'а.</p>

Microsoft Dynamics NAV 5.0 Quick Reference RU

TOTALSCAUSEDDBY	FieldNo := TOTALSCAUSEDDBY Используется для определения, какое поле вызвало выполнение операций с переменными. Может использоваться только в секциях group header и group footer.
URL	String :=Report.URL([UseNames]) Возвращает строку с полным URL отчета.
USEREQUESTFORM	[IsUseRequestForm] := USEREQUESTFORM([SetUseRequestForm]) Используется для возврата текущего свойства UseReqForm (использовать форму запроса), и для установки этого свойства в новое значение. Эта функция должна использоваться до вызова request form (формы запроса) в триггере OnInitReport. В случае его использования после вызова формы запроса ошибки не будет, но никакого эффекта от этой функции тоже не будет.

STRINGS

Существуют следующие типы текстовых данных: *BIGTEXT*, *CODE*, *TEXT*. Нормальные строковые функции не могут быть использованы с переменной *BigText*.

BIGTEXT

ADDTEXT `BigText.ADDTEXT(Variable [,Position])`
Используется для добавления текстовой строки в переменную *BigText*. Строку можно вставить в любом месте переменной, а также в конце. Для удаления содержимого переменной *BigText* используется функция *CLEAR*.

GETSUBTEXT `[RetLength] := BigText.GETSUBTEXT(Variable, Position [,Length])`
Используется для получения какой-либо части текста в переменной *BigText*.

LENGTH `Length := BigText.LENGTH`
Используется для получения длины переменной *BigText*.

READ `[Ok] := BigText.READ(InStream)`
Используется для переноса содержимого *BigText*, хранимого в таблице как *BLOB*, в переменную *BigText*.

TEXTPOS `Position := BigText.TEXTPOS(String)`
Используется для получения позиции, в которой специальная строка впервые встречается в *BigText*.

WRITE `[Ok] := BigText.WRITE(OutStream)`
Используется для переноса *BigText* в поле *BLOB* в таблице.

STRING

CONVERTSTR `NewString := CONVERTSTR(String, FromCharacters, ToCharacters)`
Используется для конвертирования символов в строке из символов *FromCharacters* в символы *ToCharacters*. Если длины *FromCharacters* и *ToCharacters* не эквивалентны, то произойдет ошибка запуска.

COPYSTR `NewString := COPYSTR(String, Position [, Length])`
Используется для копирования подстроки любой длины из определенной позиции в строке (текст или код) в новую строку. Если опустить параметр *Length* (длина), результирующая строка будет включать все символы, начиная от *Position* и до конца строки.

DELCHR `NewString := DELCHR(String [, Where] [, Which])`
Используется для удаления одного или более символов в строке.

DELSTR `NewString := DELSTR(String, Position [, Length])`
Используется для удаления подстроки внутри строки (текст или код).

FORMAT `String := FORMAT(Value [, Length] [, FormatStr/Number])`
Используется для форматирования числового значения в строковое.

INCSTR `NewString := INCSTR(String)`
Используется для увеличения положительного числа или уменьшения отрицательного внутри строки последовательно на 1.

Microsoft Dynamics NAV 5.0 Quick Reference RU

INSSTR	NewString := INSSTR(String, SubString, Position) Используется для вставки подстроки в строку.
LOWERCASE	NewString := LOWERCASE(String) Используется для конвертирования всех букв строки в подстрочные (lowercase).
MAXSTRLEN	MaxLength := MAXSTRLEN(String) Используется для возврата максимальной длины строковой переменной.
PADSTR	NewString := PADSTR(String, Length [, FillCharacter]) Используется для изменения длины строки в длину, определенную пользователем. Система выполнит функцию даже в случае уменьшения длины строки или добавления символов в конце строки.
SELECTSTR	NewString := SELECTSTR(Number, CommaString) Используется для получения подстроки из строки, разделенной запятыми.
STRCHECKSUM	CheckNumber := STRCHECKSUM(String [, WeightString] [, Modulus]) Используется для вычисления контрольной суммы для строки, содержащей номер.
STRLEN	Length := STRLEN(String) Используется для возврата длины строки.
STRPOS	Position := STRPOS(String, SubString) Используется для поиска подстроки внутри строки.
STRSUBSTNO	NewString := STRSUBSTNO(String [, Value1, ...]) Используется для перемещения полей %1, %2, %3... и #1, #2, #3... в строку со значениями, которые вы укажете в опциональных параметрах.
UPPERCASE	NewString := UPPERCASE(String) Используется для конвертирования всех букв строки в надстрочные (uppercase).

SYSTEM

ARRAY	
ARRAYLEN	Length := ARRAYLEN(Array [, Dimension]) Используется для возврата общего количества элементов в массиве или количества элементов в специальном измерении.
COMPRESSARRAY	[Count] := COMPRESSARRAY(StringArray) Используется для перемещения всех непустых строк (текста) в массиве на начало массива (компрессия). Результирующий StringArray имеет то же самое количество элементов, что и входящий массив, но пустые записи и записи, содержащие пробелы, появятся в конце массива.
COPYARRAY	COPYARRAY(NewArray, Array, Position [, Length]) Используется для копирования одного или более элементов массива в новый массив. Разрешается копировать данные только в массивах одного измерения. Для копирования массивов 2х и 3х разных измерений функцию COPYARRAY необходимо повторить.

Microsoft Dynamics NAV 5.0 Quick Reference RU

CODECOVERAGE

CODECOVERAGELOG

[IsActive] := CODECOVERAGELOG([NewIsActive])

Используется для запуска и остановки ведения лога кода. Можно также использовать ее для получения текущего статуса лога. Когда вы хотите получить полный текст кода, генерируемый при запуске Microsoft Dynamics NAV или при тестировании приложения, вы должны запустить из командной строки инструмент Code Coverage. Для запуска Microsoft Dynamics NAV с инструментом Code Coverage, введите следующую команду: "fin.exe COVERAGELOG". Для запуска Microsoft Dynamics NAV с инструментом Code Coverage и для записи лога в файл xx.xx, введите следующую команду: "fin.exe COVERAGELOG, COVERAGEFILENAME=xx.xx".

LANGUAGE

GLOBALLANGUAGE

[LanguageID] := GLOBALLANGUAGE([NewLanguageID])

Используется для установки и получения текущего глобального языка C/SIDE. LanguageID – это ID стандартного языка Windows. Виртуальная таблица Windows Language содержит список этих ID и соответствующие ей имена.

LANGUAGE

[CurrLanguage] := LANGUAGE([NewLanguage])

Используется для установки и получения текущих настроек языка для объекта (форма, отчет или dataport).

WINDOWSLANGUAGE

LanguageID := WINDOWSLANGUAGE

Используется для получения текущих настроек языка Windows.

OPERATING SYSTEM

APPLICATIONPATH [5.0]

APPLICATIONPATH

Используется для возврата пути директории, в котором установлен исполняемый файл для Microsoft Dynamics NAV. Строка с директорией заканчивается обратной косой чертой ('\') и не содержит имя исполняемого файла. В строке не может содержаться более 255 символов

COMMANDLINE

String := COMMANDLINE

Используется для возврата списка параметров, используемых для запуска Microsoft Dynamics NAV.

CONTEXTURL

String :=CONTEXTURL

Используется для возврата контекстной строки, определяющей текущий URL запущенных объектов. Примеры контекстной строки:

navision://client/run?database=filename&company=companyname

navision://client/run?server=servername&company=companyname&servertype=MSSQL

ENVIRON

String := ENVIRON(Name)

Используется для возврата строки, связанной с переменной окружения. Если переменной окружения не существует, возвращаемая строка может содержать в себе ненужные данные (мусор).

Microsoft Dynamics NAV 5.0 Quick Reference RU

GUIALLOWED	<p>[Ok] := GUIALLOWED()</p> <p>Данная функция ОС используется для проверки, разрешается ли кодом C/AL показ какой-либо информации на экране. При запуске Application Server для Microsoft Dynamics NAV, GUIALLOWED всегда возвращает FALSE, а любой вызов функций CONFIRM или dialog.OPEN, или любая попытка использовать форму или датапорт приведут к ошибке.</p>
HYPERLINK	<p>HYPERLINK(URL)</p> <p>Запускает Microsoft Internet Explorer, передавая в качестве аргумента URL. Функция HYPERLINK (гиперссылка) работает с номерами протоколов и типами файлов, утвержденными в используемой вами версии программы. Эти протоколы и типы файлов перечислены в файле .stx.</p>
OSVERSION	<p>String := OSVERSION</p> <p>Используется для возврата строки, содержащей имя и версию ОС. Эта строка сообщает тип и версию ОС. Вот некоторые типичные примеры: Windows 98 -> Windows_95_4.10; Windows NT -> Windows_NT_4.0; Windows 2000 -> Windows_NT_5.0; Windows XP -> Windows_NT_5.1; Windows 2003 -> Windows_NT_5.2.</p>
SHELL	<p>[ReturnCode] := SHELL(Name [, Param, ...])</p> <p>Используется для запуска внешних программ и команд ОС из C/AL-программ. Вы можете запускать эту функцию модально или немодально, в зависимости от того, включаете ли вы возвращаемое значение из внешней программы. Функция SHELL была разработана таким образом, чтобы каждый пользователь мог создать список доверенных программ. Этот список хранится в zip-файле пользователя.</p>
SLEEP	<p>SLEEP(Duration)</p> <p>Передаёт управление операционной системе на определенное количество времени (в миллисекундах).</p>
TEMPORARYPATH [5.0]	<p>TEMPORARYPATH</p> <p>Используется для возврата пути директории, в котором хранится временный файл для Microsoft Dynamics NAV. Эта строка заканчивается бэкслэшем ('\') и не содержит имени временного файла. Строка не может содержать более 255 символов.</p>
VARIABLE CLEAR	<p>CLEAR(Variable)</p> <p>Используется для очистки значения переменной. CLEAR также удаляет все наложенные на переменную фильтры и меняет ключ на первичный. Используйте функцию CLEARALL для очистки всех внутренних переменных, ключей и фильтров на объектах, таких как отчеты, формы, codeunit'ы и других, содержащих код C/AL. Однако заметьте, что CLEARALL не затрагивает и не изменяет значения для переменных в единичных случаях для codeunit'ов.</p>

Microsoft Dynamics NAV 5.0 Quick Reference RU

CLEARALL	<p>CLEARALL</p> <p>Используется для очистки всех внутренних переменных (за исключением переменных REC), ключей фильтров на объектах, таких как отчеты, формы, codeunit'ы и других, содержащих код C/AL. CLEARALL работает через повторяющийся вызов функции CLEAR на каждую переменную. Однако, в случае с codeunit'ами, CLEARALL работает через вызов функции CLEARALL внутри codeunit'а. Она удаляет содержимое codeunit'а, тогда как CLEAR удаляет только ссылку на codeunit.</p>
COPYSTREAM	<p>[Ok] := COPYSTREAM(OutStream, InStream)</p> <p>Используется для копирования информации, содержащейся в InStream, на OutStream.</p>
EVALUATE	<p>[Ok] := EVALUATE(Variable, String [, Number])</p> <p>Используется для оценки (evaluate) представления значения строки. Система записывает результат в переменную.</p>

VARIANT

Тип данных *variant* может содержать любые разновидности (*variant*) данных из ОСХ и объектов Automation (VT_VARIANT). Variant может содержать следующие типы данных C/AL: record (запись), file (файл), action (действие), codeunit, Automation, Boolean (булева переменная), option, integer, decimal, char, text, code, date, time, binary, DateFormula, TransactionType, InStream и OutStream. Для возврата переменных C/AL необходимо передать их в параметре ByVar (в COM называемый ByRef).

DATI2VARIANT Variant := DATI2VARIANT(Date, Time)
 Данная функция системной даты используется для создания variant, содержащий VT_DATE.

VARIANT2DATE Date := VARIANT2DATE(Variant)
 Используется для возврата даты из VT_DATE.

VARIANT2TIME Time := VARIANT2TIME(Variant)
 Используется для возврата времени из VT_DATE.

Для определения того, содержит ли *variant* некоторые типы переменных, используются следующие функции:

ISACTION	Ok := Variant.ISACTION
ISAUTOMATION	Ok := Variant.ISAUTOMATION
ISBINARY	Ok := Variant.ISBINARY
ISBOOLEAN	Ok := Variant.ISBOOLEAN
ISCHAR	Ok := Variant.ISCHAR
ISCODE	Ok := Variant.ISCODE
ISCODEUNIT	Ok := Variant.ISCODEUNIT
ISDATE	Ok := Variant.ISDATE
ISDATEFORMULA	Ok := Variant.ISDATEFORMULA
ISDECIMAL	Ok := Variant.ISDECIMAL
ISFILE	Ok := Variant.ISFILE
ISINSTREAM	Ok := Variant.ISINSTREAM
ISINTEGER	Ok := Variant.ISINTEGER
ISOPTION	Ok := Variant.ISOPTION
ISOUTSTREAM	Ok := Variant.ISOUTSTREAM
ISRECORD	Ok := Variant.ISRECORD
ISTEXT	Ok := Variant.ISTEXT
ISTIME	Ok := Variant.ISTIME
ISTRANSACTIONTYPE	Ok := Variant.ISTRANSACTIONTYPE

VIRTUAL TABLES

2000000001	Object	2000000029	System Object	2000000052	Windows Group Member
2000000002	User	2000000037	Performance	2000000053	Windows Access Control
2000000003	Member Of	2000000038	AllObj	2000000054	Windows Login
2000000004	User Role	2000000039	Printer	2000000055	SID - Account ID
2000000005	Permission	2000000040	License Information	2000000056	User SID
2000000006	Company	2000000041	Field	2000000058	AllObjWithCaption
2000000007	Date	2000000042	OLE Control	2000000059	Breakpoint
2000000009	Session	2000000043	License Permission	2000000061	User Menu Level
2000000010	Database File	2000000044	Permission Range	2000000063	Key
2000000020	Drive	2000000045	Windows Language	2000000065	Send-To Program [5.0]
2000000022	File	2000000046	Automation Server	2000000066	Style Sheet [5.0]
2000000024	Monitor	2000000049	Code Coverage	2000000067	User Default Style Sheet [5.0]
2000000026	Integer	2000000050	Windows Object	2000000068	Record Link [5.0]
2000000028	Table Information	2000000051	Service Connection Point	2000000203	Database Key Groups

XMLPORT

Объект XMLport концептуально связан с dataport'ом; XMLport'ы также используются для импорта и экспорта данных, но в формате XML. XMLport'ы делают процесс обмена данными в XML между системами более простым и прямолинейным. Для работы с XMLport'ом нужны только базовые знания XML и создавать документы XML, используя внешние программы, просто нет необходимости.

BREAK	BREAK Используется для выхода из цикла или триггера dataport'а, отчета или XMLport'а.
QUIT	QUIT Прекращает процесс работы dataport'а, отчета или XMLport'а.
SKIP	SKIP Пропускает текущий шаг итерации при работе dataport'а, отчета или XMLport'а.
XMLport.EXPORT	[[Ok]] := EXPORT(Number, OutStream[, Record]) Используется для создания потока данных XML (документа XML) и пересылки его в пункт назначения.
XMLport.IMPORT	[[Ok]] := IMPORT(Number, InStream [, ResponseOutStream]) Используется для чтения и разбора входящего потока данных XML (документа XML).