Динамические списки

Используется для отображения информации из любых таблиц независимо от их типа.

Похоже, динамический список формируется на сервере и затем представление передается на клиент. Поэтому использование именно динамического списка, а не например таблицы значений, должно быть осознанно. Динамический список предоставляет встроенные опции (стандартные команды, переходы для выделенных объектов), однако может сильно тормозить работу в случаях например часто меняющейся выборки из большого объема данных. Иногда эффективнее один раз загрузить на клиент данные и организовать фильтрацию на клиенте, без постоянной нагрузки на сервер и каналы связи.

Шаги использования динамического списка:

1 Наполнение данными

Данные могут быть получены из основной таблицы или через произвольный запрос. Способ получения данных настраивается в свойствах динамического списка, галочка Произвольный запрос.

Для обоих вариантов существует настройка Отбор (какие данные попадут в итоговый список), результат его работы пересекается с настройками запроса, поэтому требует углубленного понимания. Скорее всего, такая херня возникла из-за попытки сделать сложную вещь доступной прогерам-мышкотыкам (как и все в 1С), но потом пришли к выводу, что потенциал у данного типа данных большой, реализовать все возможные варианты через удобный графический интерфейс нельзя, а часть интерфейса создана, пусть будет. Понять и простить.

	Основная таблица	Произвольный запрос
Набор данных	Поля одной таблицы	Поля любого запроса
Интерактивная настройка (через отбор)	Настройка в конфигураторе повторяет настройку отображения в пользовательском интерфейсе. Ручное указание конкретного параметра либо общие типы (пустое, больше/меньше, ...)	Нельзя изменить запрос, остальное так же
Программная настройка параметров, включая параметры из реквизитов	Нет	Есть
Встроенный функционал редактирования списка	Да	Только если указана опция "Основная таблица".

Можно сделать вывод: настройка динамического списка через отбор конфигуратора актуальна для простейших задач (убрать пустые, ограничить по фиксированному значению, ...). В остальных случаях эффективнее работать через фильтрацию в параметрах запроса. Поэтому настройку отбора через вкладки с получением данных из основной таблицы рассмотрим в разделе Настройка отображения, далее будем рассматривать фильтрацию через произвольный запрос.

Произвольный запрос. Выберем следующие данные:

```
ВЫБРАТЬ
	ДокументПоступлениеТоваров.Ссылка,
	ДокументПоступлениеТоваров.Номер,
	ДокументПоступлениеТоваров.Дата,
	ДокументПоступлениеТоваров.Фирма,
	ДокументПоступлениеТоваров.Филиал,
	ДокументПоступлениеТоваров.ПереченьТоваров
ИЗ
	Документ.ПоступлениеТоваров КАК ДокументПоступлениеТоваров
ГДЕ
	ДокументПоступлениеТоваров.Дата < &НужнаяДата
```

Для параметров необходимо задать значение перед использованием. Это настраивается в модуле формы в процедуре «ПриСозданииНаСервере».

```
&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
	ДокументыПоступленийТоваров.Параметры.УстановитьЗначениеПараметра("НужнаяДата",
	ТекущаяДата());
```

```
КонецПроцедуры
```

ДокументыПоступленийТоваров - имя реквизита Динамический список.

2 Настройка отображения

Для всех разделов.
Несколько правил объединяются по способу И, правила можно объединять в группы, а для групп настраивать способ И, ИЛИ, НЕ.
В рамках одного пользователя сохраняется последняя настроенная группа правил.
Можно создавать преднастроенные наборы правил.

Отбор. Задается фиксированная настройка динамического списка. Виды сравнения зависят от типа данных поля, бывают в частности фиксированные (равно, больше, меньше), в списке, заполнено, в группе и обратные операции.

Порядок. Задается поле и тип сортировки. Несколько правил применяются последовательно, в рамках одинакового первого правила сортируется по следующему и т д.

Условное оформление. Может быть несколько правил, для каждого правила параметры оформления, условия и оформляемые поля. Параметры оформления: цвет фона, цвет текста, шрифт, положение текста, выделение отрицательного, шаблон текста, видимость, доступность. Т е здесь, как и в правилах отбора, можно даже ограничить видимость элементов. Сравнение способов ограничения видимости:

	Запрос	Отбор	Условное оформление
Точка фильтрации	На сервере	На сервере	На клиенте
Простота	Сложно	Просто	Просто
Скорость создания фильтра	Медленно	Быстро	Быстро
Настраиваемость	Динамическая	Статическая	Статическая
Объем передаваемых данных	Только попавшие в фильтр	Только попавшие в фильтр	Все
Изменение пользователем	То что разрешено разработчиком	Да	Да

Группировка. Несколько правил применяются последовательно, в рамках одинакового первого правила группируется по следующему и т д.

3 Интерактивное взаимодействие

В случае установленной Основной таблицы доступен стандартный набор команд (создание, удаление, изменение).

```
Элементы.элСписок.ТекущаяСтрока = НоменклатураСсылка; //получение и изменение текущей строки,
Сообщить(Элементы.элСписок.ТекущаяСтрока.Цена);// данные заново получаются из базы
Сообщить(Элементы.элСписок.ТекущиеДанные.Цена);// данные получаются из существующих данных


//перебор строк динамического цикла
&НаСервере
Процедура ЦиклПоСтрокамДинамическогоСписка()
    ТЗ_ДС = СписокВТЗ();
    Для Каждого Стр Из ТЗ_ДС Цикл
        // .........
    КонецЦикла;
КонецПроцедуры
```

5 Обновление данных

Сама по себе информация в списках не обновляется. Необходимо два компонента: процедура, генерирующая информацию о факте изменения данных и способ распространения информации об изменении по открытым формам (метод обновления). Стандартного общего события при обновлении данных например у справочника не нашел. Поэтому для реализации задачи поддержки актуальности чувствительных данных в открытых формах возможны следующие варианты:

1. Отказаться от стандартного механизма работы с данными, переписать процедуры обновления в отдельные формы и добавить оповещение об обновлении (см. ниже). При небольшом количестве точек изменения это сработает, но поддержка муторная. И если кто-то из разработчиков случайно забудет о необходимости использования только специальных процедур и добавит случайно прямое изменение данных - появятся фантомные неактуальные данные. А еще - см. блок "Нае*али, но понять и простить"
2. Периодический контроль на уровне таблицы средствами языка 1С и оповещения в случае изменения. Упрощает задачу с точки зрения контроля точек изменения данных, но хорошо нагружает сервер. К тому же, это будет обновление с задержкой (минимум в минуту для файловой базы). Если таблицы объемные, данные меняются относительно нечасто, то постоянное хеширование сильно замедлит работу.
3. Вариации на тему генерации события изменения на уровне базы данных, дополнительных сервисов на чем-нибудь. Возможно с добавлением endpoint в 1С. Например, клиент-серверный вариант, БД Postgresql (для mssql похоже все гораздо сложнее). В 1С поднимается web сервис, при обращении на который генерируются оповещения по открытым клиентским формам. На postgresql устанавливается расширение plpython3u, настраивается триггер на изменение нужных таблиц и за счет этого генерируется REST запрос при изменении данных. Решение на случай "ПИЗДЕЦ КАК НАДО СОБЫТИЙ ДОХЕРА В РАЗНЫХ ТОЧКАХ ПОЛЬЗЫ НЕ МОГУТ САМИ ПОСТОЯННО ОБНОВЛЯТЬ, НА СЕРВЕР НЕТ ДЕНЕГ ЗАТО ЕСТЬ ДЕНЬГИ НА РАЗРАБОВ".

Решение сложно сопровождаемое, зависящее от имен нужных таблиц, требующее разработчиков на других языках. Очень дорогое.

4. Периодический опрос сервера из формы. Нагрузка растет произведением от количества клиентов, количества контролируемых таблиц, их объема и минимального периода обновления. Если добавить возможности периодического регистра сведений, то нагрузка перестанет зависеть от количества клиентов. Возможно эту задачу вынести на отдельный сервер, организовав например кластер БД, скрипт по пересчету на другом сервере и еще какое-нибудь извращение. Но все равно обновление с задержкой, от 1 минимального периода обновления. Самый допотопный способ, но при некоторых условиях имеет право на использование. Дополнение: когда это писал, реально думал что это древняя технология и нужно просто узнать, как это реализуется в 1С. Но... Похоже это единственный рабочий способ,
5. Есть отдельный продукт 1С:Предприятие - Сервер взаимодействия, но нужно тестировать.

Но в целом, 1С даже близко не позиционируется как система реального времени / мгновенного обновления / совместного редактирования, так что особых претензий к отсутствию данных технологий быть не может. Возможно, стоит пересмотреть формулировку задачи или вынести данный функционал совсем за пределы 1С.

В перечисленных способах используется термин "Оповещение об обновлении". Методы оповещения представлены ниже. Метод ОповеститьОбИзменении работает только для динамических списков, да и то только для списков с указанной основной таблицей. Метод Оповестить универсальный.

Здесь возникает парадокс: обычно начинает использоваться дополнительная технология в связи с упрощением решения нужной задачи. Однако если эта технология ведет к дополнительным трудностям, то стоит сопоставить получаемые преимущества, возникающее усложнение и степень решения задачи.

Методы обновления

стр. 576 книги Разработка интерфейсов.

Метод ОповеститьОбИзменении() В него передается ссылка на объект (или ключ записи), об изменении которого нужно оповестить формы. Он уведомит все динамические списки, расположенные в созданных на клиенте формах, об изменении этого объекта, и они обновят свои данные. Но этот метод не обновит те динамические списки, у которых не задана основная таблица. Преимущество: процесс автоматический, без изменения форм. Но этот метод можно использовать только из клиента. Как следствие, списки в другой сессии / у другого пользователя не обновляются.

```
//ДобавитьЭлементНаСервере() - функция, возвращающая ссылку на новый элемент
//в какой-то форме
СсылкаНаНовыйЭлемент = ДобавитьЭлементНаСервере();
```

```
ОповеститьОбИзменении(СсылкаНаНовыйЭлемент);
```

Метод Оповестить() Нужно добавить код обработки оповещений в формах, в которых нужно что-то обновлять. Метод
глобального контекста Оповестить() отсылает оповещение всем созданным (не обязательно открытым) формам. В форме в обработчике события ОбработкаОповещения() можно обработать это сообщение и выполнить нужную модификацию формы.

И один хер у другого пользователя не обновляется!

```
&НаКлиенте
Процедура ОбщееОповещение(Команда)
	Элем = СоздатьСОповещениемНаСервере();
	Оповестить("ДобавлениеЭлемента");
КонецПроцедуры

&НаКлиенте
Процедура ОбработкаОповещения(ИмяСобытия, Параметр, Источник)
	Если ИмяСобытия = "ДобавлениеЭлемента" Тогда
		Элементы.ИзСправочника.Обновить();
	КонецЕсли;
КонецПроцедуры
```

Нае*али, но понять и простить.

У 1С нет стандартных технологии а-ля longpool, только метод Обновить.

Revision #7
Created 6 February 2025 14:29:14 by Admin
Updated 8 February 2025 14:42:54 by Admin