

ГРУППОВАЯ РАЗРАБОТКА

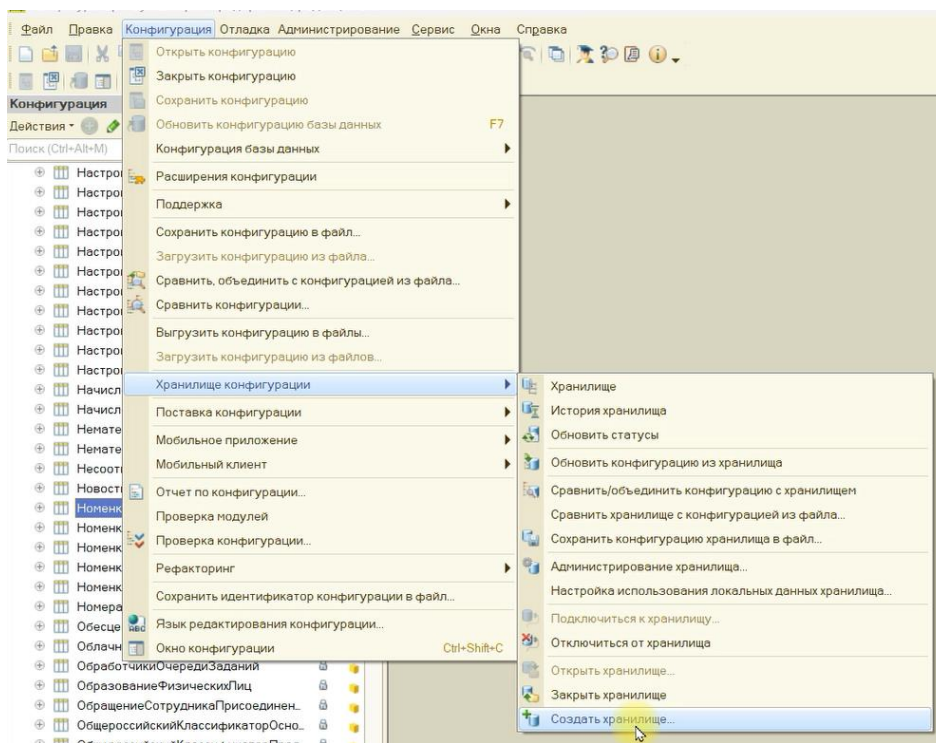
В этом модуле мы познакомимся с таким понятием как групповая разработка. Посмотрим то, каким образом несколько разработчиков могут одновременно вести разработку по изменению одной и той же конфигурации.

Рассмотрим, что такое хранилище конфигурации и для чего оно нужно.

Если одна база открыта в конфигураторе и другой пользователь попытается тоже зайти в конфигуратор. Он увидит сообщение об ошибке. То есть второй разработчик уже не может зайти в конфигуратор, когда один там что-то делает.

Поэтому придумали такую систему как хранилище конфигурации. При помощи хранилища конфигурации несколько разработчиков могут одновременно дорабатывать одну и ту же конфигурацию.

Каким образом? Создается хранилище конфигурации через конфигуратор.

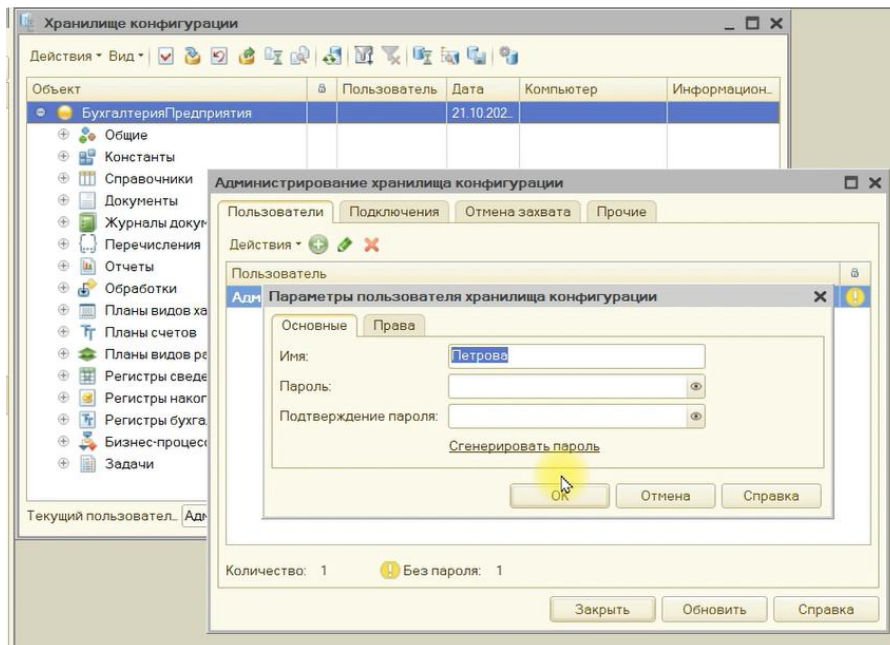


Что администратор будет делать после того, как создал хранилище?

Администратор осуществляет формирование списка пользователей, имеющих доступ к хранилищу, может просматривать список пользователей, подключенных к хранилищу, а также освобождать объекты конфигурации от захвата.

Он может это все делать в меню Конфигурация, Хранилище конфигурации. Создавать пользователей администратор может в меню Действия, Администрирование хранилища.

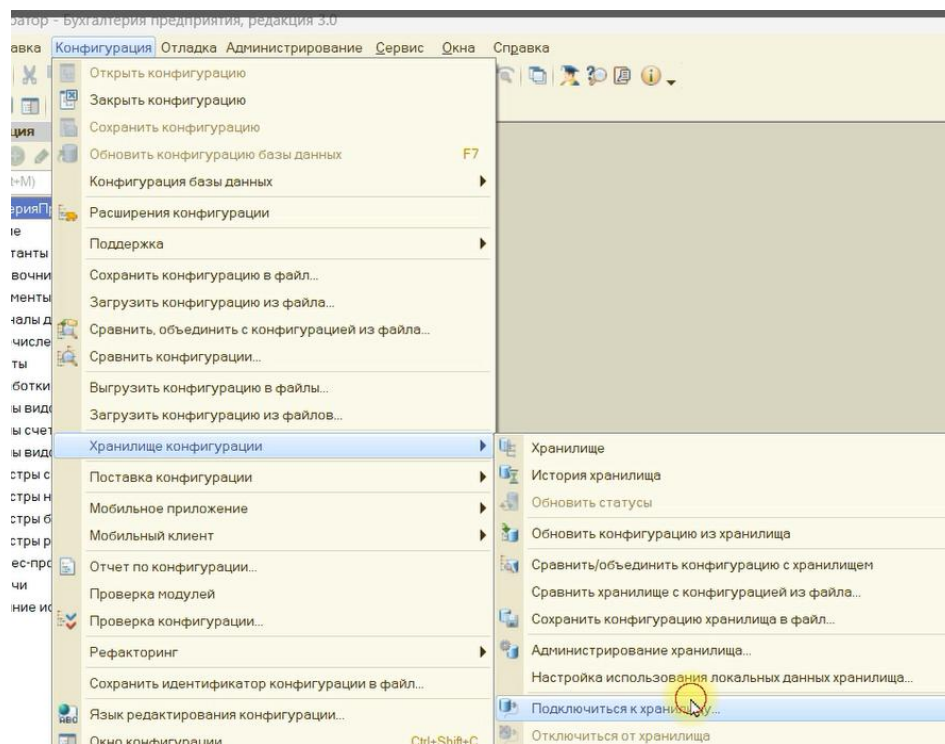
Допустим, Ваш руководитель хочет, чтобы вы тоже подключились к хранилищу. Для этого он задает здесь нового пользователя, задает ему какое-то имя пользователя и пароль. И задает вам какие-то права.



Для того, чтобы пользователь смог подключиться к хранилищу, ему понадобится пользователь, пароль и еще путь к хранилищу.

И теперь, зная путь к этой папке хранилища, зная пользователя и пароль, новый разработчик может подключиться к данному хранилищу. Он открывает другую базу. Какую-то свою, тестовую базу, заходит в ее конфигуратор. Под каким пользователем он заходит, это вообще никак не влияет на работу с хранилищем.

И теперь зайдя в свою, абсолютно другую базу, но с такой же конфигурацией, он может подключиться к хранилищу. Для этого ему нужно зайти в меню конфигурация, хранилище конфигурации, подключиться к хранилищу.



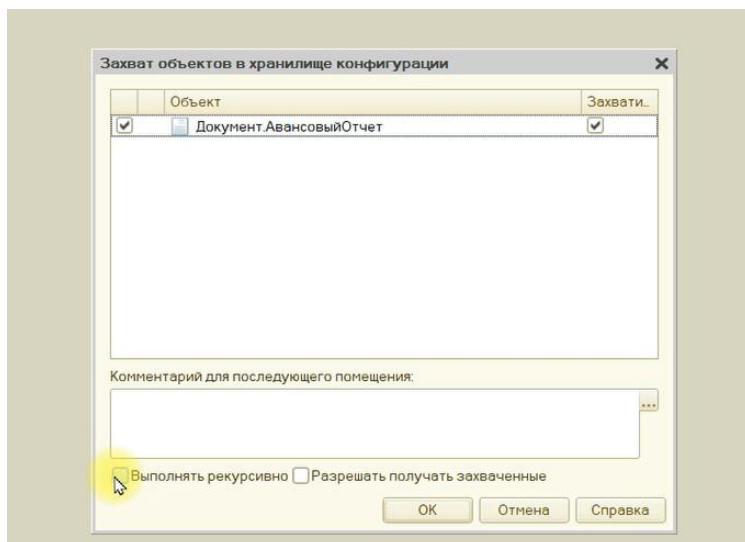
Вы должны понимать, что ваша конфигурация при подключении к хранилищу будет заменена конфигурацией из хранилища.

То есть какие-то ваши доработки, если были в этой базе, они затрутсся.

Что нужно первым делом всегда делать, когда вы заходите в базу, которая у вас подключена к хранилищу. Первым делом всегда нужно нажать конфигурация хранилища конфигурации, обновить конфигурацию из хранилища. Это позволит вам подтянуть в свою конфигурацию все изменения, сделанные за последнее время другими разработчиками в хранилище конфигурации. То есть вы получаете актуальную версию конфигурации со всеми изменениями.

Как происходит разработка?

Вы видите окно вашей конфигурации со всеми справочниками, документами, обработками и так далее. И когда вам нужно работать с каким-то определенным объектом, вы его находите. Для того, чтобы его менять, вам нужно первым делом нажать на этот объект и захватить его в хранилище.



Здесь у вас есть возможность выполнять этот захват рекурсивно, либо не ставить эту галочку выполнять рекурсивно. Что это обозначает?

Если мы поставим выполнять рекурсивно, то у нас будет помимо самого объекта захвачены еще и его формы и макеты.

Для операции захвата есть такие единицы. Это сам объект, либо каждая его форма, либо каждый его макет. Это все можно захватывать в отдельности.

Нельзя в отдельности захватить табличную часть или нельзя в отдельности захватить какой-то реквизит. Их можно, реквизиты и табличные части, захватить только вместе с объектом.

Какие вообще операции с хранилищем может выполнять разработчик.

У него есть такие варианты. Захватить в хранилище, поместить в хранилище, отменить захват в хранилище и получить из хранилища.

Получить из хранилища мы нажимаем, когда мы сомневаемся, что у нас в конфигурации актуальный этот объект. То есть мы Думаем, что возможно, кто-то мог его поменять. И у нас сейчас в конфигурации не актуален на этот объект. Мы нажимаем «Получить из хранилища».

Естественно, лучше всего перед разработкой каждого объекта нажимать «Получить из хранилища».

«Захватить в хранилище». Мы нажимаем «Захватить в хранилище» и только после этого мы можем этот объект менять. можем как-то его дорабатывать.

Если мы передумали дорабатывать наш объект, мы можем просто нажать отменить захват хранилища и все наши изменения просто пропадут и никак не отразятся на хранилище конфигурации.

Другим разработчикам наши изменения не попадут, если мы нажмем отменить захват хранилища.

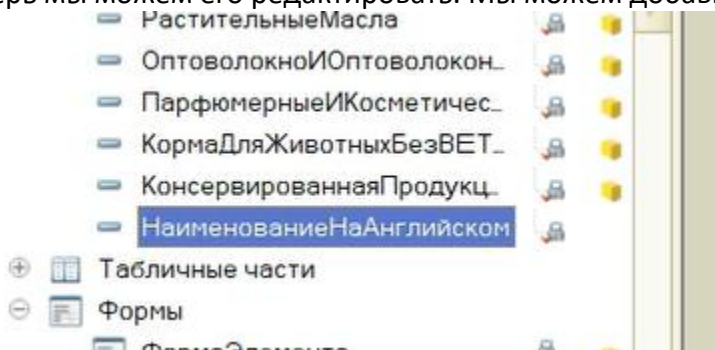
И последняя команда - это поместить в хранилище. Это когда вы сделали свою доработку, вы ее потестировали на своей тестовой базе, пока мы не нажмем поместить в хранилище, наши изменения не попадут к другим разработчикам. Наши изменения не попадут в хранилище.

Давайте попробуем сделать теперь какую-то мини-задачку. Допустим, у нас такая задачка. Нам нужно добавить в номенклатуру новый реквизит наименование на английском и поменять для реквизита артикул длину. Раньше у него была длина 25 символов, а теперь нам нужно сделать, чтобы у реквизита артикул длина была 30 символов.

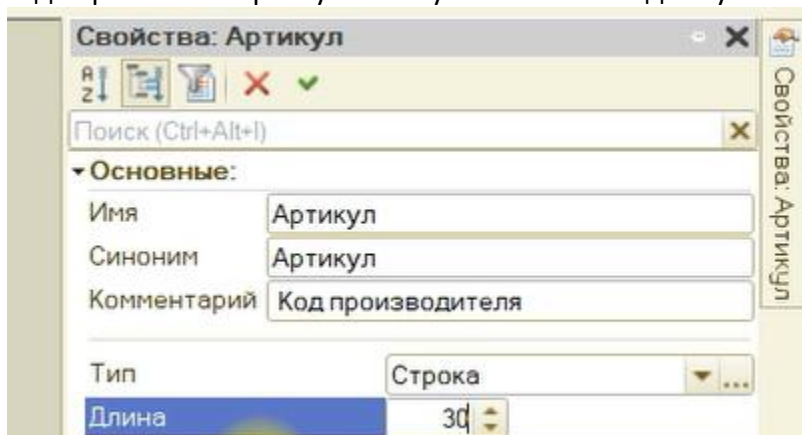
Мы идем и находим нужный нам справочник. Захватываем его.

Сейчас мы уже понимаем, что работать мы хотим только с реквизитами номенклатуры, поэтому нам не нужно захватывать его рекурсивно, чтобы не захватились формы, чтобы формы остались доступны другим разработчикам. Поэтому мы захватываем справочник не рекурсивно.

Теперь мы можем его редактировать. Мы можем добавить новый реквизит.



И для реквизита артикул нам нужно поменять длину. Мы меняем длину.



Все, мы выполнили все наши изменения. И теперь помещаем нашу номенклатуру с изменениями в хранилище. Нажимаем поместить в хранилище.

Для выполнения операции с хранилищем необходимо сохранить конфигурацию, продолжить, да.

Все, операция завершена и наши изменения помещены в хранилище. Теперь когда другие разработчики обновят конфигурацию из хранилища, они увидят новый реквизит они увидят, что у артикула другая длина.

Чтобы добавить новый объект, допустим, если мы захотим добавить новый справочник, у нас эта функция будет неактивна, пока мы не захватим корень конфигурации.

Мы должны захватить в хранилище корень конфигурации. Выполнять рекурсивно нам не нужно ставить. Просто нужно захватить корень конфигурации. Тогда мы можем добавлять новые объекты.

Допустим, добавляем мы новый справочник. Добавили мы новый справочник. И, допустим, мы хотим теперь здесь очень долго что-то думать над нашим справочником, какие у него будут реквизиты, какие у него будут формы, какие табличные части. Нам не обязательно теперь думать над этим справочником, пока у нас захвачен корень конфигурации.

То есть пока мы свой справочник дорабатываем, больше никто из разработчиков вообще не может добавлять никакие объекты в конфигурацию, потому что вами захвачен корень.

После того как вы добавили новый объект, вы можете отпустить этот захват в хранилище, только для начала включите новый справочник в какую-то подсистему. Иначе выйдет ошибка при обновлении конфигурации и у вас и у других разработчиков.

Потом можно корень конфигурации поместить в хранилище. Вместе с нашим новым справочником.

То есть все, мы просто создали объект и отпустили корень конфигурации для того, чтобы другие разработчики уже могли добавлять в это время какие-то другие объекты, а мы теперь спокойно работаем над нашим новым справочником, добавляем ему реквизиты, табличные части, формы, макеты и так далее.

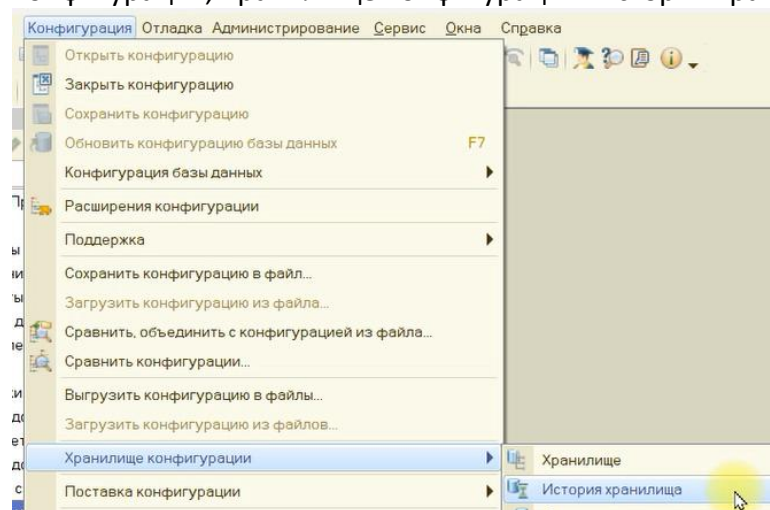
Точно так же и с другими объектами. Обратите внимание на регистры накопления. Если вы добавите таким образом, например, регистр накопления без регистраторов или без ресурсов и поместите эти изменения в хранилище, вот такой пустой регистр без регистраторов и без ресурсов, то и у вас и у других разработчиков при попытке обновить свою конфигурацию базы данных будет критическая ошибка, которую пропустить нельзя и она не даст обновить конфигурацию ни вам, ни другим разработчикам.

Поэтому вот когда вы создаете регистры накопления, нельзя помещать их пустыми с одним только названием в хранилище, нужно обязательно создать им ресурсы и указать, что является для них регистратором.

Проверяйте перед тем, как поместить в хранилище, при помощи конфигурация обновить конфигурацию базы данных, обновляется ли конфигурация. Только после этого можно уже отпускать корень конфигурации, когда вы добавляли какой-то новый объект.

Можем ли мы посмотреть, кто вообще сейчас с чем работает и что делает с хранилищем конфигурации?

Да, у конфигурации существует история хранилища. Мы можем зайти в нее через конфигурация, Хранилище конфигурации. История хранилища.



Здесь показано какой пользователь и что делал.

Также, что еще вам нужно знать о работе с хранилищем.

То, что в любой момент мы можем наши объекты сравнить с объектом хранилища, либо сравнить выборочно с объектом хранилища.

А также вы можете в любой момент сравнить свою конфигурацию с конфигурацией хранилища. Как это сделать? Вы должны щелкнуть на корень и нажать сравнить и объединить конфигурацию с хранилищем.

И у вас откроется окно сравнения и объединения конфигураций, где вы можете смотреть, чем отличаются ваши конфигурации от конфигурации хранилища.

Должна ли быть рабочая база подключена к хранилищу? Это совсем не обязательно и даже это не нужно.

Рабочая база не должна быть подключена к хранилищу. Хранилище это инструмент исключительно для разработчиков, в нем разработчики разрабатывают на своих тестовых базах конфигурацию, а уже переносить сделанные доработки нужно в рабочую базу аккуратно через меню конфигурация сравнить, объединить с конфигурацией из файла, это должен быть выгруженный из тестовой базы файл cf. Актуальный, полученный только что из хранилища cf файл.

Можно ли, если у вас используется работа с хранилищем, пойти и сделать сразу какую-то доработку в рабочей базе, не делая ее сначала в хранилище?

Нельзя, если у вас принято то, что вы работаете с хранилищем, вы в любом случае должны сделать пойти сначала даже эту мини-доработку в хранилище, поместить в хранилище, потом уже сохранить эту конфигурацию в файл cf и через этот cf файл переносить свою доработку. Почему?

Потому что если вы не сделаете сначала в хранилище, а сделаете сразу в рабочей базе, то другой разработчик посчитает, что конфигурация актуальная в хранилище, пойдет переносить уже свои доработки при помощи своего cf файла в рабочую базу и увидит, что различия не только там, где он дорабатывал, а еще и, допустим, в вашей номенклатуре, которую вы, не поместив в хранилище, пошли и сделали только на рабочей базе.

Ему придется разбираться, почему сейчас разница не только в его доработках, а вот еще здесь в какой-то номенклатуре.

То есть просто вы этим самым тратите время чужое.

Если принято работать с использованием хранилища, вы должны любые доработки, даже самые мелкие, сначала помещать в хранилище а потом уже переносить на рабочую базу.

Источник: <https://t.me/blogpro1c>