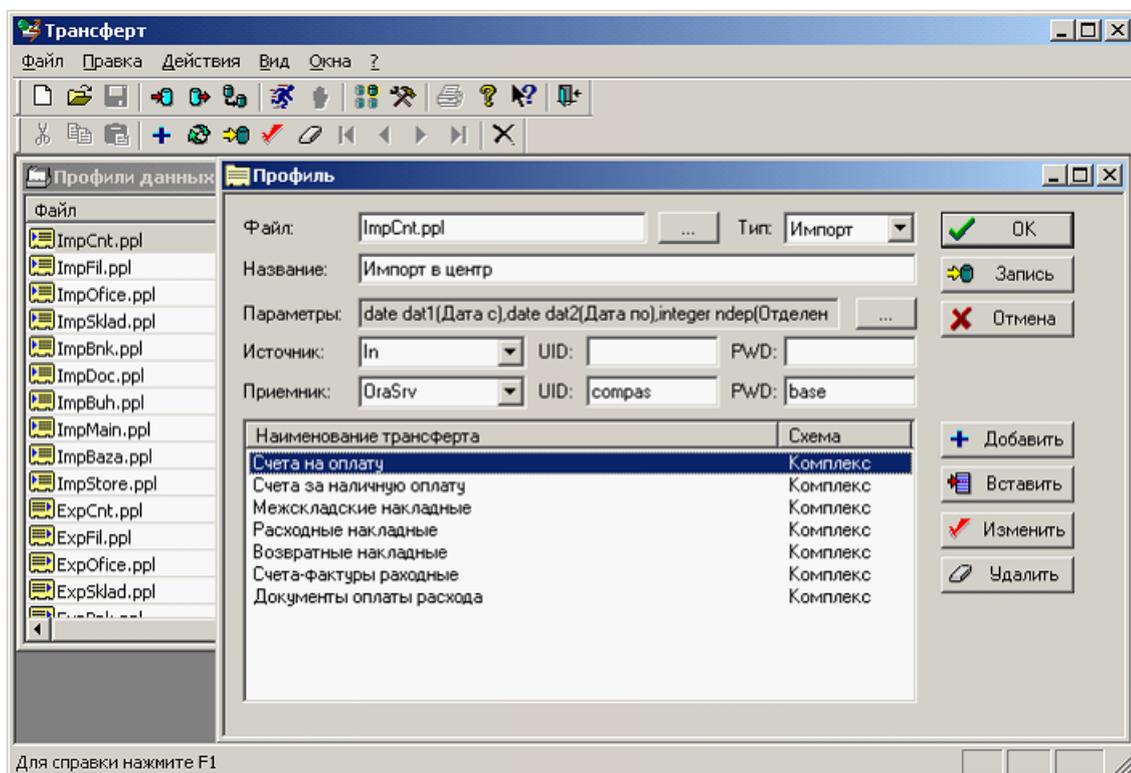


# 1 Введение



Трансферт – это унифицированное настраиваемое приложение для автоматической синхронизации (репликации) данных различных баз данных (БД). Программа написана на языке Microsoft Visual C++ с использованием ODBC API, что позволяет добиться наивысшей производительности при работе с большими массивами данных. Унификация программы Трансферт обеспечивается стандартизованным интерфейсом к различным СУБД, предоставляемым драйверами ODBC. Широкий выбор существующих драйверов ODBC делает программу Трансферт универсальной для работы как с файловыми БД (dBase, Access), так и с реляционными СУБД (Oracle, Sybase, Microsoft SQL). При работе Трансферт использует внутренний формат представления данных, что делает его независимым от специфических типов данных, присущих различным базам данных. Наличие встроенного языка описания данных позволяет настраивать программу для выполнения любых задач по передаче или конвертированию информации.



## 2 Характеристики

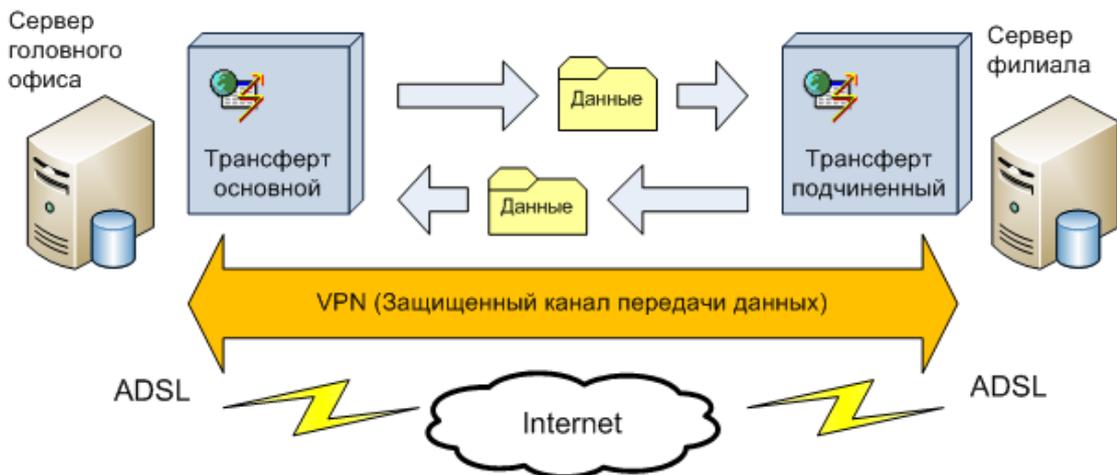
- работа с любыми базами данных через интерфейс ODBC
- высокая скорость обработки данных
- поддержка унифицированного языка запросов SQL
- автоматическое преобразование типов данных при передаче
- встроенный гибкий язык описания структуры данных (профилей)
- поддержка внешних параметров профиля данных
- удобный редактор для определения профиля данных
- возможность вызова хранимых процедур БД при передаче данных
- событийно-управляемый сценарий обработки данных с помощью механизма триггеров
- возможность обработки связанных данных в одном блоке передачи по типу основная/

подчиненная таблица

- встроенный конвертор текстовых данных из ANSI в OEM и обратно
- набор опций для запуска из командной строки
- возможность ведения файла протокола
- возможность трассировки выполнения операторов SQL
- возможность автоматического формирования файлов сообщений с результатами операции
- возможность настройки автоматического файлового обмена данными с удаленным узлом по принципу основной/подчиненный
- внутренний упаковщик файлов в формате LZW
- многоязыковая поддержка

### 3 Области применения

1. Программу удобно использовать как утилиту для быстрой одноразовой передачи или конвертирования данных из одной базы данных в другую, которые при этом не обязательно должны быть однородными. Например, совершенно легко может быть выполнена передача данных между базами данных Access и Oracle. Все, что для этого нужно – это описать структуру передаваемых данных (таблиц) между источником и приемником. При этом можно воспользоваться свойством автоматического создания таблиц в базе данных приемника. Данные могут передаваться как напрямую, так и в режиме сравнения, то есть добавляться будут только отсутствующие записи в таблице приемника.
2. Можно использовать Трансферт как мощное средство, обеспечивающее обмен данными для бухгалтерско-управленческой программы между центральным офисом и удаленными филиалами. При этом может быть задействован мощный механизм комплексной обработки данных для передачи документов (как неделимого целого, состоящего из связанных данных). Для автоматизации обработки данных при импорте информации возможно использование событийных сценариев с помощью описания специальных триггеров. Также огромным преимуществом является возможность вызова хранимых процедур БД при передаче данных.
3. Трансферт может служить как передаточный механизм для обеспечения автоматического непрерывного обмена данными между двумя удаленными базами данных. Специальная настройка обмена данными позволяет организовать автоматическое выполнение сеансов передачи данных через определенные интервалы времени. Программа при этом должна быть постоянно запущена (быть резидентной). Для удобства пользователя предусмотрено сворачивание программы в System Tray. Для минимизации размера передаваемых данных в программе предусмотрен встроенный упаковщик в формате LZW.



## 4 Описание работы

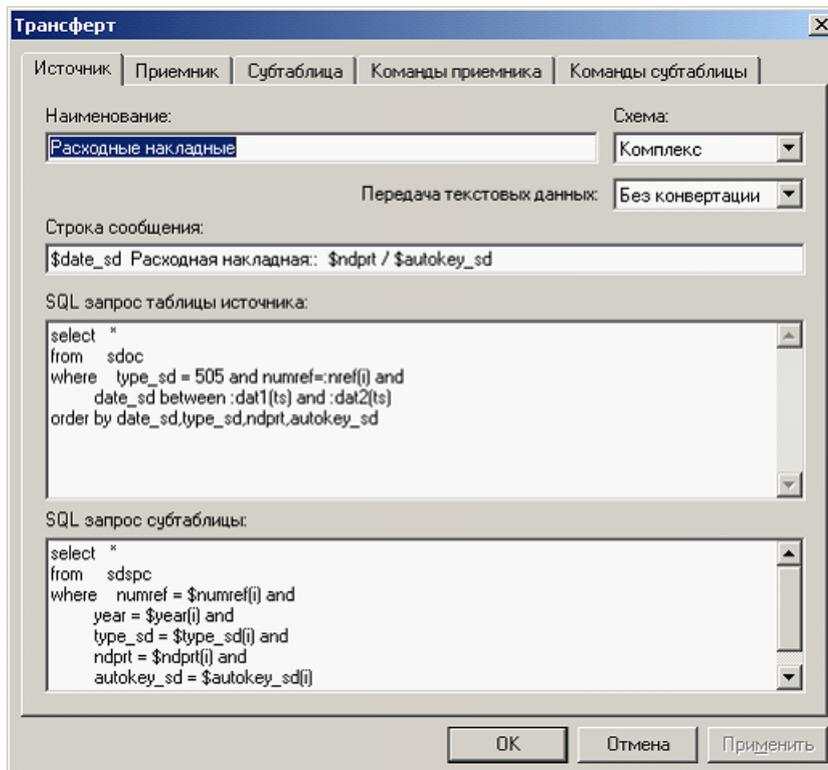
Для работы Трансферта необходимо наличие драйверов ODBC для базы данных источника и приемника. Далее необходимо описать структуру данных (профиль) и сохранить ее в файле. Для этого можно воспользоваться любым текстовым редактором или встроенным диалоговым редактором, что не требует знания синтаксиса языка. Можно ввести в профиль список параметров, значения которых могут быть установлены непосредственно в момент выполнения (например, диапазон дат). Затем созданный файл профиля должен быть зарегистрирован в списке текущих профилей, чтобы быть доступным для выполнения. С этого момента операцию передачи данных можно выполнить, выбрав соответствующий профиль. Можно установить выбранный профиль как профиль по умолчанию для операции импорта или экспорта данных. При этом инициализация нужного профиля будет выполняться автоматически.

Существуют несколько режимов выполнения Трансферта:

1. **Командная строка.** Программу можно запустить из командной строки, указав необходимый набор параметров.
2. **Ручной режим.** Программа запускается вручную, выбирается необходимый профиль и выполняется.
3. **Автоматический режим.** Программа постоянно загружена в памяти компьютера. Для удобства программа может быть свернута в системный Tray. Текущий статус программы отображается специальной иконкой. Требуемые операции по передаче данных выполняются через определенные промежутки времени.

При определении профиля можно задать один из следующих алгоритмов работы:

- Создание таблицы
- Добавление данных
- Замещение данных
- Полное сравнение



## 5 Команды обработки данных

Команды используются для описания событийно-управляемого сценария выполнения некоторых связанных действий при передаче данных. Запуск команд привязан к выполнению операторов SQL в таблице приемника. В качестве команд могут выступать операторы SQL (Insert, Update, Delete) или вызовы хранимых процедур БД.

**Трансферт**

Источник | Приемник | Субтаблица | Команды приемника | Команды субтаблицы

Таблица команд: Добавить Вставить Удалить

№	Команда	Событие	Поле
1	update sdoc set flag=1 where type_sd=\$type_sd and numref=\$numref ar ndprt=\$ndprt and autokey_sd=\$autokey_sd	CallExec	
2	update sdoc set flag=0 where type_sd=\$type_sd and numref=\$numref ar ndprt=\$ndprt and autokey_sd=\$autokey_sd	CallBack	
3	call BAL_MOD(\$numref,\$year,\$cli1_ndprt,\$cli1_n \$code_currency,\$sumf_sd,^0,\$date_sd,\$flag)	OnInsert	
4	call BAL_MOD(\$numref,\$year,\$cli1_ndprt,\$cli1_n \$code_currency,\$sumf_sd,#sumf_sd,\$date_sd,#l	OnUpdate	
5	call BAL_MOD(#numref,#year,#cli1_ndprt,#cli1_r #code_currency,^0,#sumf_sd,#date_sd,#flag)	OnDelete	
6	call BAL_MOD(\$numref,\$year,\$cli1_ndprt,\$cli1_n \$code_currency,^0,\$sumf_sd,\$date_sd,\$flag)	ReInsert	
7	call BAL_MOD(\$numref,\$year,\$cli1_ndprt,\$cli1_n \$code_currency,#sumf_sd,\$sumf_sd,\$date_sd,#l	ReUpdate	
8	call BAL_MOD(#numref,#year,#cli1_ndprt,#cli1_r #code_currency,#sumf_sd,^0,#date_sd,#flag)	ReDelete	
9	update card_spc set date_card=\$date_sd where numref=\$numref and year=\$year and sklac sds.numref=\$numref and sds.year=\$year an	OnChange	date_sd

OK Cancel Apply