**3.39. Подключаемые команды**

Подсистема «Подключаемые команды» предоставляет программный интерфейс для вывода различных команд в формах, списках и журналах  программы. В частности, её программный интерфейс используется для вывода команд подсистем [Печать](https://its.1c.ru/db/content/bsp301doc/src/3.38.%20печать.htm?_=1542292879#_печать), [Заполнение объектов](https://its.1c.ru/db/content/bsp301doc/src/3.20.%20заполнение%20объектов.htm?_=1542292879#_заполнение_объектов), [Дополнительные отчеты и обработки](https://its.1c.ru/db/content/bsp301doc/src/3.16.%20дополнительные%20отчеты%20и%20обработки.htm?_=1542292879#_дополнительные_отчеты_и) и [Варианты отчетов](https://its.1c.ru/db/content/bsp301doc/src/3.7.%20валюты.htm?_=1542292879#_варианты_отчетов) (подменю Печать, Отчеты, Заполнить и др.).

**Настройка**

**Подключить формы объектов программы**

Для подключения форм объектов программы, в которых требуется выводить подменю Печать, Отчеты или Заполнить, необходимо:

● В процедуре ПриСозданииНаСервере (обработчик события формы) вставить вызов по шаблону:

// СтандартныеПодсистемы.ПодключаемыеКоманды

ПодключаемыеКоманды.ПриСозданииНаСервере(ЭтотОбъект);

// Конец СтандартныеПодсистемы.ПодключаемыеКоманды

● В модуле формы вставить процедуры (обработчики команд):

// СтандартныеПодсистемы.ПодключаемыеКоманды

&НаКлиенте

Процедура Подключаемый\_ВыполнитьКоманду(Команда)

ПодключаемыеКомандыКлиент.ВыполнитьКоманду(ЭтотОбъект, Команда, <ОбъектИлиТаблицаФормы>);

КонецПроцедуры

&НаСервере

Процедура Подключаемый\_ВыполнитьКомандуНаСервере(Контекст, Результат)

ПодключаемыеКоманды.ВыполнитьКоманду(ЭтотОбъект, Контекст, <ОбъектИлиТаблицаФормы>, Результат);

КонецПроцедуры

&НаКлиенте

Процедура Подключаемый\_ОбновитьКоманды()

ПодключаемыеКомандыКлиентСервер.ОбновитьКоманды(ЭтотОбъект, <ОбъектИлиТаблицаФормы>);

КонецПроцедуры

// Конец СтандартныеПодсистемы.ПодключаемыеКоманды

● Если форма является формой объекта, тогда:

● В параметре <ОбъектИлиТаблицаФормы> следует передавать реквизит формы типа ДанныеФормыСтруктура. Например:

ПодключаемыеКомандыКлиент.ВыполнитьКоманду(ЭтотОбъект, Команда, Объект);

● В обработчике ПриЧтенииНаСервере формы следует вставить вызов по шаблону:

&НаСервере

Процедура ПриЧтенииНаСервере(ТекущийОбъект)

// СтандартныеПодсистемы.ПодключаемыеКоманды

ПодключаемыеКомандыКлиентСервер.ОбновитьКоманды(ЭтотОбъект, <ОбъектИлиТаблицаФормы>);

// Конец СтандартныеПодсистемы.ПодключаемыеКоманды

КонецПроцедуры

● В обработчике ПриОткрытии формы следует вставить вызов по шаблону:

&НаКлиенте

Процедура ПриОткрытии(Отказ)

// СтандартныеПодсистемы.ПодключаемыеКоманды

ПодключаемыеКомандыКлиент.НачатьОбновлениеКоманд(ЭтотОбъект);

// Конец СтандартныеПодсистемы.ПодключаемыеКоманды

КонецПроцедуры

● Также рекомендуется вызывать процедуры обновления видимости команд после изменения ключевых реквизитов (значения которых могут быть использованы в условиях видимости команд):

● ПодключаемыеКомандыКлиент.НачатьОбновлениеКоманд - для обновления видимости команд на клиенте (подключает обработчик ожидания Подключаемый\_ОбновитьКоманды).

● ПодключаемыеКомандыКлиентСервер.ОбновитьКоманды - для обновления видимости команд на сервере (используется в случае, если в процессе вызова уже делается серверный вызов).

● Если форма является формой списка, тогда:

● В параметре <ОбъектИлиТаблицаФормы> следует передавать элемент формы типа ТаблицаФормы, связанный с динамическим списком. Например:

ПодключаемыеКомандыКлиент.ВыполнитьКоманду(ЭтотОбъект, Команда, Элементы.Список);

● В обработчике ПриАктивизацииСтроки таблицы формы следует вставить вызов по шаблону:

&НаКлиенте

Процедура <ИмяТаблицы>ПриАктивизацииСтроки(Элемент)

// СтандартныеПодсистемы.ПодключаемыеКоманды

ПодключаемыеКомандыКлиент.НачатьОбновлениеКоманд(ЭтотОбъект);

// Конец СтандартныеПодсистемы.ПодключаемыеКоманды

КонецПроцедуры

Внимание. Поле Ссылка динамического списка формы должно быть доступно в обработчике команды. Для этого необходимо в свойствах поля Ссылка, вложенного в реквизит формы типа ДинамическийСписок, включить флажок Использовать всегда.

● Опционально. Добавить реквизит формы ПараметрыПодключаемыхКоманд произвольного типа. Это позволит не создавать реквизит формы динамически, что ускорит время открытия формы.

**Настройка прав доступа пользователей**

Для настройки прав доступа дополнительных действий не требуется.

Таблица 3.78.

| № | Роли и их назначение |
| --- | --- |
| 1. | БазовыеПраваБСП или БазовыеПраваВнешнихПользователейБСП (из подсистемы «Базовая функциональность»):  Просмотр информации. |

**Использование при разработке конфигурации**

**Подключение отчетов и обработок к механизмам конфигурации**

В отчетах и обработках конфигурации и расширений можно разрабатывать дополнительные команды объектов конфигурации для вывода в подменю Печать, Отчеты или Заполнить, а также настраивать варианты отчета и переопределять поведение общей формы отчета. Для этого:

● Отчет или обработка должна входить в состав подсистемы ПодключаемыеОтчетыИОбработки.

● В модуле менеджера должна быть определена процедура ПриОпределенииНастроек по шаблону:

#Область ПрограммныйИнтерфейс

// Определяет состав программного интерфейса для интеграции с конфигурацией.

//

// Параметры:

//   Настройки - Структура - Настройки интеграции этого объекта.

//       Описание см. в модуле ПодключаемыеКомандыПереопределяемый,

//       комментарий к процедуре ПриОпределенииСоставаНастроекПодключаемыхОбъектов,

//       раздел "Параметры процедуры ПриОпределенииНастроек"

//

Процедура ПриОпределенииНастроек(Настройки) Экспорт

КонецПроцедуры

#КонецОбласти

● Структура параметра Настройки процедуры ПриОпределенииНастроек обработок:

Таблица 3.79. Настройки обработок, включенных в состав подсистемы ПодключаемыеОтчетыИОбработки.

| Настройка | Тип | Описание |
| --- | --- | --- |
| Размещение | Массив из ОбъектМетаданных | Объекты, к которым подключен этот объект.  Процедуры ДобавитьКомандыПечати, ДобавитьКомандыЗаполнения и ДобавитьКомандыОтчетов вызываются только из форм тех объектов метаданных, которые указаны в этом параметре. |
| ДобавитьКомандыПечати | Булево | Если Истина то в модуле менеджера следует определить процедуру по шаблону:  // Заполняет список команд печати.  //  // Параметры:  //   КомандыПечати - ТаблицаЗначений - Подробнее см. в УправлениеПечатью.СоздатьКоллекциюКомандПечати().  //  Процедура ДобавитьКомандыПечати(КомандыПечати) Экспорт  // Код по добавлению команд печати.  КонецПроцедуры |
| ДобавитьКомандыЗаполнения | Булево | Если Истина то в модуле менеджера следует определить процедуру по шаблону:  // Определяет список команд заполнения.  //  // Параметры:  //   КомандыЗаполнения - ТаблицаЗначений - Таблица с командами заполнения. Для изменения.  //       См. описание 1 параметра процедуры ЗаполнениеОбъектовПереопределяемый.ПередДобавлениемКомандЗаполнения().  //   Параметры - Структура - Вспомогательные параметры. Для чтения.  //       См. описание 2 параметра процедуры ЗаполнениеОбъектовПереопределяемый.ПередДобавлениемКомандЗаполнения().  //  Процедура ДобавитьКомандыЗаполнения(КомандыЗаполнения, Параметры) Экспорт  // Код по добавлению команд заполнения.  КонецПроцедуры |

● Структура параметра Настройки процедуры ПриОпределенииНастроек отчетов:

Таблица 3.80. Настройки отчетов, включенных в состав подсистемы ПодключаемыеОтчетыИОбработки.

| Настройка | Тип | Описание |
| --- | --- | --- |
| Размещение | Массив из ОбъектМетаданных | Объекты, к которым подключен этот объект.  Процедуры ДобавитьКомандыПечати, ДобавитьКомандыЗаполнения и ДобавитьКомандыОтчетов вызываются только из форм тех объектов метаданных, которые указаны в этом параметре. |
| ДобавитьКомандыПечати | Булево | Если Истина то в модуле менеджера следует определить процедуру по шаблону:  // Заполняет список команд печати.  //  // Параметры:  //   КомандыПечати - ТаблицаЗначений - Подробнее см. в УправлениеПечатью.СоздатьКоллекциюКомандПечати().  //  Процедура ДобавитьКомандыПечати(КомандыПечати) Экспорт  // Код по добавлению команд печати.  КонецПроцедуры |
| ДобавитьКомандыЗаполнения | Булево | Если Истина то в модуле менеджера следует определить процедуру по шаблону:  // Определяет список команд заполнения.  //  // Параметры:  //   КомандыЗаполнения - ТаблицаЗначений - Таблица с командами заполнения. Для изменения.  //       См. описание 1 параметра процедуры ЗаполнениеОбъектовПереопределяемый.ПередДобавлениемКомандЗаполнения().  //   Параметры - Структура - Вспомогательные параметры. Для чтения.  //       См. описание 2 параметра процедуры ЗаполнениеОбъектовПереопределяемый.ПередДобавлениемКомандЗаполнения().  //  Процедура ДобавитьКомандыЗаполнения(КомандыЗаполнения, Параметры) Экспорт  // Код по добавлению команд заполнения.  КонецПроцедуры |
| ДобавитьКомандыОтчетов | Булево | Если Истина то в модуле менеджера следует определить процедуру по шаблону:  // Определяет список команд отчетов.  //  // Параметры:  //   КомандыОтчетов - ТаблицаЗначений - Таблица с командами отчетов. Для изменения.  //       См. описание 1 параметра процедуры ВариантыОтчетовПереопределяемый.ПередДобавлениемКомандОтчетов().  //   Параметры - Структура - Вспомогательные параметры. Для чтения.  //       См. описание 2 параметра процедуры ВариантыОтчетовПереопределяемый.ПередДобавлениемКомандОтчетов().  //  Процедура ДобавитьКомандыОтчетов(КомандыОтчетов, Параметры) Экспорт  // Код по добавлению команд отчетов.  КонецПроцедуры |
| НастроитьВариантыОтчета | Булево | Если Истина то в модуле менеджера следует определить процедуру по шаблону:  // Настройки размещения в панели отчетов.  //  // Параметры:  //   Настройки - Коллекция - Передается "как есть" из ВариантыОтчетовПереопределяемый.НастроитьВариантыОтчетов().  //       Может использоваться для получения настроек варианта этого отчета при помощи функции ВариантыОтчетов.ОписаниеВарианта().  //   НастройкиОтчета - СтрокаДереваЗначений - Настройки этого отчета,  //       уже сформированные при помощи функции ВариантыОтчетов.ОписаниеОтчета() и готовые к изменению.  //       См. "Свойства для изменения" процедуры ВариантыОтчетовПереопределяемый.НастроитьВариантыОтчетов().  //  // Описание:  //   См. ВариантыОтчетовПереопределяемый.НастроитьВариантыОтчетов().  //  // Вспомогательные методы:  // НастройкиВарианта = ВариантыОтчетов.ОписаниеВарианта(Настройки, НастройкиОтчета, "<ИмяВарианта>");  // ВариантыОтчетов.УстановитьРежимВыводаВПанеляхОтчетов(Настройки, НастройкиОтчета, Истина/Ложь);  //  // Примеры:  //  //  1. Установка описания варианта.  // НастройкиВарианта = ВариантыОтчетов.ОписаниеВарианта(Настройки, НастройкиОтчета, "<ИмяВарианта>");  // НастройкиВарианта.Описание = НСтр("ru = '<Описание>'");  //  //  2. Отключение возможности выбора варианта отчета и вывода в интерфейсах подсистемы (в частности - в панелях отчетов).  // НастройкиВарианта = ВариантыОтчетов.ОписаниеВарианта(Настройки, НастройкиОтчета, "<ИмяВарианта>");  // НастройкиВарианта.Включен = Ложь;  //  Процедура НастроитьВариантыОтчета(Настройки, НастройкиОтчета) Экспорт  // Код процедуры.  КонецПроцедуры |
| ОпределитьНастройкиФормы | Булево | Отчет имеет программный интерфейс для тесной интеграции с формой отчета, в том числе может переопределять некоторые настройки формы и подписываться на ее события.  Если Истина и отчет подключен к общей форме ФормаОтчета, тогда в модуле объекта отчета следует определить процедуру по шаблону:  // Настройки общей формы отчета подсистемы "Варианты отчетов".  //  // Параметры:  //   Форма - УправляемаяФорма, Неопределено - Форма отчета или форма настроек отчета.  //       Неопределено когда вызов без контекста.  //   КлючВарианта - Строка, Неопределено - Имя предопределенного  //       или уникальный идентификатор пользовательского варианта отчета.  //       Неопределено когда вызов без контекста.  //   Настройки - Структура - см. возвращаемое значение  //       ОтчетыКлиентСервер.ПолучитьНастройкиОтчетаПоУмолчанию().  //  Процедура ОпределитьНастройкиФормы(Форма, КлючВарианта, Настройки) Экспорт  // Код процедуры.  КонецПроцедуры |

См. также примеры в демонстрационной базе: в объектах конфигурации и расширений, включенных в состав подсистемы ПодключаемыеОтчетыИОбработки.

**Расширение видов подключаемых команд**

В стандартной поставке предусмотрены следующие виды подключаемых команд: печатные формы, отчеты и команды заполнения. Однако есть возможность добавлять и собственные виды подключаемых команд.

Для этого в модуле ПодключаемыеКомандыПереопределяемый:

● В процедуре ПриОпределенииВидовПодключаемыхКоманд определить собственные виды подключаемых команд и задать умолчания для вывода подменю с командами этих видов. Например:

Процедура ПриОпределенииВидовПодключаемыхКоманд(ВидыПодключаемыхКоманд) Экспорт

Вид = ВидыПодключаемыхКоманд.Добавить();

Вид.Имя         = "Мотиваторы";

Вид.ИмяПодменю  = "ПодменюМотиваторов";

Вид.Заголовок   = НСтр("ru = 'Мотиваторы'");

КонецПроцедуры

● В процедуре ПриОпределенииСоставаНастроекПодключаемыхОбъектов описать ключ (например, типа Булево), который будет служить маркером того, что объект поставляет команды данного вида. Например:

Процедура ПриОпределенииСоставаНастроекПодключаемыхОбъектов(НастройкиПрограммногоИнтерфейса) Экспорт

Настройка = НастройкиПрограммногоИнтерфейса.Добавить();

Настройка.Ключ          = "ДобавитьМотиваторы";

Настройка.ОписаниеТипов = Новый ОписаниеТипов("Булево");

Настройка.ВидыПодключаемыхОбъектов = "Обработка";

КонецПроцедуры

● В процедуре ПриОпределенииКомандПодключенныхКОбъекту добавить команды данного вида, которые соответствуют переданному контексту. Например:

Процедура ПриОпределенииКомандПодключенныхКОбъекту(НастройкиФормы, Источники, ПодключенныеОтчетыИОбработки, Команды) Экспорт

<МодульПодсистемы>.ПриОпределенииКомандПодключенныхКОбъекту(НастройкиФормы, Источники, ПодключенныеОтчетыИОбработки, Команды);

КонецПроцедуры

● Описать в документации как разрабатывать команды данного вида.

Более детальные примеры см. в модулях и документации подсистемы [Заполнение объектов](https://its.1c.ru/db/content/bsp301doc/src/3.20.%20заполнение%20объектов.htm?_=1542292879#_заполнение_объектов).

**Настройка обмена данными**

Для настройки обмена данными никаких действий не требуется, так как подсистема не предоставляет собственных данных.