**3.16. Дополнительные отчеты и обработки**

Подсистема «Дополнительные отчеты и обработки» предназначена для подключения и использования дополнительных (внешних) отчетов и обработок к информационной базе в режиме 1С:Предприятие.

В зависимости от назначения дополнительные отчеты и обработки бывают глобальными, если используются в целом для конфигурации, либо назначаемыми, если используются с конкретными типами объектов информационной базы. Назначаемые обработки делятся по своей функциональности на четыре типа: для заполнения объекта, печатные формы, для создания на основании и отчеты.

**Настройка**

Для использования подсистемы в конфигурации необходимо поместить справочник ДополнительныеОтчетыИОбработки в командный интерфейс администратора. Затем выполнить шаги, описанные ниже.

**Настройка назначаемых дополнительных отчетов и обработок**

Необходимо принять решение по поводу состава объектов метаданных, к которым должны подключаться команды дополнительных отчетов и обработок. Например, это могут быть все справочники и документы конфигурации. Эти объекты метаданных необходимо указать в составе свойстве Тип определяемого типа ОбъектСДополнительнымиКомандами.

Для каждого объекта метаданных необходимо внести изменения во все его формы объекта и списка, встроив подсистему "[Подключаемые команды](https://its.1c.ru/db/content/bsp301doc/src/3.39.%20подключаемые%20команды.htm?_=1542292879#_поиск_и_удаление)".

Состав объектов метаданных, к которым можно подключать дополнительные печатные формы, определяется при внедрении подсистемы "[Печать](https://its.1c.ru/db/content/bsp301doc/src/3.38.%20печать.htm?_=1542292879#_печать)".

**Настройка глобальных дополнительных отчетов и обработок**

1. Необходимо принять решение по поводу места размещения в командном интерфейсе конфигурации команд для вызова глобальных отчетов и обработок. Рекомендуется размещать такие команды в подсистемах верхнего уровня, участвующих в формировании командного интерфейса (разделы глобального командного интерфейса).

2. Затем для каждого места размещения глобальных дополнительных обработок создать общую команду вида ДополнительныеОбработки<место\_размещения>, включить ее в состав функциональной опции ИспользоватьДополнительныеОтчетыИОбработки и добавить в нее код по шаблону:

&НаКлиенте

Процедура ОбработкаКоманды(ПараметрКоманды, ПараметрыВыполненияКоманды)

ДополнительныеОтчетыИОбработкиКлиент.ОткрытьФормуКомандДополнительныхОтчетовИОбработок(

ПараметрКоманды,

ПараметрыВыполненияКоманды,

ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработкиДополнительнаяОбработка(),

"<Имя раздела>");

КонецПроцедуры

где <имя раздела> – имя раздела командного интерфейса, в котором размещена команда. Для получения имени рабочего стола следует использовать функцию ИдентификаторРабочегоСтола общего модуля ДополнительныеОтчетыИОбработкиКлиентСервер.

В качестве примера такой команды см. общую команду ДополнительныеОбработкиАдминистрирование в демонстрационной конфигурации.

3. Аналогичным образом для каждого места размещения глобальных дополнительных отчетов надо создать общую команду вида ДополнительныеОтчеты<место\_размещения>, включить ее в состав функциональной опции ИспользоватьДополнительныеОтчетыИОбработки и вставить в нее следующий код:

&НаКлиенте

Процедура ОбработкаКоманды(ПараметрКоманды, ПараметрыВыполненияКоманды)

ДополнительныеОтчетыИОбработкиКлиент.ОткрытьФормуКомандДополнительныхОтчетовИОбработок(

ПараметрКоманды,

ПараметрыВыполненияКоманды,

ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработкиДополнительныйОтчет(),

"<Имя раздела>");

КонецПроцедуры

где <имя раздела> – имя раздела командного интерфейса, в котором размещена команда. Для получения имени рабочего стола следует использовать функцию ИдентификаторРабочегоСтола общего модуля ДополнительныеОтчетыИОбработкиКлиентСервер.

В качестве примера такой команды см. общую команду ДополнительныеОтчетыАдминистрирование в демонстрационной конфигурации.

4. В переопределяемом модуле ДополнительныеОтчетыИОбработкиПереопределяемый в процедурах ОпределитьРазделыСДополнительнымиОбработками и ОпределитьРазделыСДополнительнымиОтчетами перечислить в виде объектов метаданных разделы, в которых были размещены общие команды, созданные на предыдущих шагах. См. пример кода в демонстрационной конфигурации.

**Настройка необходимости проведения документов перед формированием внешних печатных форм**

По умолчанию перед формированием любой внешней печатной формы проверяется проведенность печатаемых объектов-документов, и если найдется хотя бы один непроведенный документ, то пользователю будет выдано соответствующее предложение его провести. Если пользователь отказывается это делать, печать не выполняется.

Такую проверку можно отключить при внедрении подсистемы. Для этого необходимо в общем модуле ДополнительныеОтчетыИОбработкиКлиентПереопределяемый в процедуре ПередВыполнениемКомандыПечатиВнешнейПечатнойФормы установить параметру СтандартнаяОбработка значение Ложь:

Процедура ПередВыполнениемКомандыПечатиВнешнейПечатнойФормы(ПечатаемыеОбъекты, СтандартнаяОбработка) Экспорт

СтандартнаяОбработка = Ложь;

КонецПроцедуры

**Размещение в командном интерфейсе**

Если в конфигурации не используется подсистема «Настройки программы», то в рабочем месте администратора программы необходимо разместить:

* ● константу ИспользоватьДополнительныеОтчетыИОбработки. При переключении значения константы регулируется возможность использования в информационной базе механизмов подсистемы. Если конфигурация адаптируется для использования в модели сервиса, отображать команду в форме настроек команды нужно только в локальном режиме работы – в модели сервиса управление значением константы выполняется централизованно из менеджера сервиса;
* ● команду открытия списка справочника ДополнительныеОтчетыИОбработки.

См. пример в форме ПечатныеФормыОтчетыИОбработки обработки ПанельАдминистрированияБСП.

**Особые случаи внедрения подсистемы**

* ● [Внедрение подсистем «Дополнительные отчеты и обработки» и «Управление доступом»](https://its.1c.ru/db/content/bsp301doc/src/3.63.%20особые%20случаи%20внедрения%20подсистем.htm?_=1542292879#_внедрение_подсистем_«дополнительные).

**Настройка прав доступа пользователей**

Для настройки прав доступа пользователей к данным подсистемы «Дополнительные отчеты и обработки» следует использовать роли, указанные ниже.

Таблица 3.31.

| № | Роли и их назначение |
| --- | --- |
| 1. | ДобавлениеИзменениеДополнительныхОтчетовИОбработок  Добавление дополнительных обработок и отчетов в информационную базу; изменение параметров публикации и объектов назначения обработок и отчетов |
| 2. | ЧтениеДополнительныхОтчетовИОбработок  Возможность запуска внешних обработок.  При совместном использовании с подсистемой «Управление доступом» для ограничения доступа к отдельным дополнительным отчетам и обработкам используется вид доступа ДополнительныеОтчетыИОбработки |

Дополнительно следует создать вспомогательные роли или использовать подходящие роли, существующие в конфигурации, для обеспечения доступа к данным, которые не относятся к подсистеме «Дополнительные отчеты и обработки», но требуются для работы с ней.

Таблица 3.32.

| № | Вспомогательные роли и их назначение |
| --- | --- |
| 1. 1. | Новая или существующая роль  Роль с установленным правом просмотра общих команд доступа к глобальным отчетам и обработкам (созданных на шагах 2 и 3 раздела настройки глобальных дополнительных отчетов и обработок). Назначается пользователям вместе с ролью ЧтениеДополнительныхОтчетовИОбработок |

Примеры настройки прав доступа пользователей приведены ниже.

Таблица 3.33.

| № | Группа пользователей и ее функции | Состав ролей |
| --- | --- | --- |
| 1. 1. | Администратор дополнительных отчетов и обработок.  Имеет возможность добавлять новые обработки в конфигурацию, публиковать их для общего использования, задавать объекты назначения и использовать внешние обработки в конфигурации | ● БазовыеПраваБСП (из подсистемы «Базовая функциональность»),  ● ЗапускТонкогоКлиента (из подсистемы «Базовая функциональность»),  ● ДобавлениеИзменениеДополнительныхОтчетовИОбработок |
| 1. 2. | Пользователь с правом выполнения дополнительных отчетов и обработок.  Имеет возможность запускать обработки на исполнение | ● БазовыеПраваБСП или БазовыеПраваВнешнихПользователейБСП (из подсистемы «Базовая функциональность»),  ● ЗапускТонкогоКлиента (из подсистемы «Базовая функциональность»),  ● ЧтениеДополнительныхОтчетовИОбработок,  ● <ЧтениеГлобальныхДополнительныхОтчетовИОбработок> |

**Использование при разработке конфигурации**

**Создание нового отчета или обработки**

Дополнительные отчеты и обработки могут быть глобальными (виды «ДополнительнаяОбработка» и «ДополнительныйОтчет») и назначаемыми (виды «ЗаполнениеОбъекта», «Отчет», «ПечатнаяФорма» и «СозданиеСвязанныхОбъектов»). Глобальные команды размещаются в глобальном командном интерфейсе. Назначаемые команды используются вместе с объектами информационной базы и размещаются в контексте тех объектов, для которых они предназначаются. Примеры назначаемых обработок см. в справочнике Демо: Контрагенты и в документе Демо: Счета на оплату покупателям в демонстрационной конфигурации. Общую информацию о работе с внешними отчетами и обработками можно найти в разделе [5.12.2. «Внешние отчеты и обработки»](https://its.1c.ru/db/v83doc/bookmark/dev/TI000000332) документации к платформе.

В общем виде последовательность создания дополнительной обработки (отчета) следующая:

1. Создать внешнюю обработку (отчет).

2. Регистрация дополнительной обработки в информационной базе происходит на основании сведений, которые поставляет сама обработка. Эти сведения должны возвращаться в виде структуры в функции СведенияОВнешнейОбработке модуля обработки. Поля структуры имеют следующее назначение:

Таблица 3.34.

| Ключ | Содержание |
| --- | --- |
| Вид | Строка, вид обработки, один из возможных: «ДополнительнаяОбработка», «ДополнительныйОтчет», «ЗаполнениеОбъекта», «Отчет», «ПечатнаяФорма» или «СозданиеСвязанныхОбъектов» |
| Назначение | Массив строк имен объектов метаданных в формате: <ИмяКлассаОбъектаМетаданного>.[ \* | <ИмяОбъектаМетаданных>].  Например, «Документ.СчетЗаказ» или «Справочник.\*». Примечание: параметр имеет смысл только для назначаемых обработок, для глобальных – не используется |
| Наименование | Наименование обработки, которым будет заполнено наименование элемента справочника, краткое описание для идентификации обработки администратором |
| Версия | Версия обработки в формате <старший номер>.<младший номер> используется при загрузке обработок в информационную базу. Например, 1.0 |
| БезопасныйРежим | Принимает значение Истина или Ложь в зависимости от того, требуется устанавливать или отключать безопасный режим при исполнении обработки (отчета). Если Истина, обработка (отчет) будет запущена в безопасном режиме. При этом, в частности, не гарантируется корректная работа дополнительных обработок под пользователями без полных прав в тех случаях, когда обработка вызывает код конфигурации, который не рассчитан на работу в безопасном режиме (в том числе неявно, когда срабатывает код модуля объекта и подписок на события при записи данных).  Более подробно о безопасном режиме см. документацию к платформе «1С:Предприятие» |
| Информация | Краткая информация по обработке, описание обработки |
| Команды | Команды, поставляемые обработкой. Таблица значений с колонками:  ● Представление – представление команды в пользовательском интерфейсе.  ● Идентификатор – идентификатор команды; любая строка, уникальная в пределах данной обработки (отчета)\*.  ● Использование – вариант запуска команды:  ● ОткрытиеФормы – открыть форму обработки;  ● ВызовКлиентскогоМетода – вызвать клиентскую экспортную процедуру из модуля формы обработки;  ● ВызовСерверногоМетода – вызвать серверную экспортную процедуру из модуля объекта обработки;  ● СценарийВБезопасномРежиме – вызвать серверную экспортную функцию из модуля объекта для формирования сценария выполнения (специальный вариант запуска для использования программного интерфейса, расширяющего доступные разработчику дополнительной обработки операции, при выполнении дополнительной обработки в безопасном режиме);  ● ЗаполнениеФормы – из модуля формы объекта вызвать серверную процедуру обработки для заполнения данных формы без записи объекта в базу данных.  ● ПоказыватьОповещение – Истина, если требуется показать оповещение при начале и при завершении работы обработки. Например, при запуске обработки без открытия формы.  ● Модификатор – дополнительный модификатор команды. Используется для дополнительных обработок печатных форм на основе табличных макетов. Для таких команд должен содержать строку ПечатьMXL (см. пример в демонстрационной конфигурации).  ● ЗаменяемыеКоманды – используется для дополнительных обработок печатных форм, содержит идентификаторы стандартных команд печати (строкой, через запятую), которые заменяются обработкой |
| ВерсияБСП | Минимальная версия БСП, на которую рассчитывает код дополнительной обработки. Должна быть задана не ниже 1.2.1.4.  Номер версии БСП задается в формате «РР.ПП.ВВ.СС» (РР – старший номер редакции; ПП – младший номер редакции; ВВ – номер версии; СС – номер сборки) |
| Разрешения | Массив разрешений, предоставляемых дополнительной обработке при использовании программного интерфейса, расширяющего доступные разработчику дополнительной обработки операции, при выполнении дополнительной обработки в безопасном режиме. Пример использования разрешений см. в обработке  \_ДемоЗагрузкаНоменклатурыИзПрайсЛистаПрофилиБезопасности демонстрационной конфигурации. |
| ОпределитьНастройкиФормы | Булево. Используется для дополнительных отчетов, подключенных к подсистеме «Варианты отчетов» и использующих общую форму отчета (подробнее см. шаг 3).  Когда Истина, то дополнительный отчет имеет программный интерфейс для тесной интеграции с формой отчета, в том числе может переопределять некоторые настройки формы и подписываться на ее события. В этом случае 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 |

\* Для обработок с печатными формами на основе макета табличного документа должен содержать список макетов, на основе которых нужно получить печатную форму (см. описание параметра ИменаМакетов процедуры УправлениеПечатьюКлиент.ВыполнитьКомандуПечати в разделе «[Печать](https://its.1c.ru/db/content/bsp301doc/src/3.38.%20печать.htm?_=1542292879#_печать)»).

Пример реализации функции СведенияОВнешнейОбработке с использованием программного интерфейса:

Функция СведенияОВнешнейОбработке() Экспорт

ПараметрыРегистрации = ДополнительныеОтчетыИОбработки.СведенияОВнешнейОбработке("2.2.2.1");

ПараметрыРегистрации.Вид = ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработки<...>();

ПараметрыРегистрации.Версия = "...";

Команда = ПараметрыРегистрации.Команды.Добавить();

Команда.Представление = НСтр("ru = '<Представление команды>'");

Команда.Идентификатор = "<Имя команды>";

Команда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКоманды<...>();

Команда.ПоказыватьОповещение = <Истина/Ложь>;

Возврат ПараметрыРегистрации;

КонецФункции

В этом примере также демонстрируется возможность автоматического заполнения параметров Наименование и Информация из метаданных внешней обработки (берутся синоним и комментарий соответственно).

Пример реализации функции СведенияОВнешнейОбработке с ручным определением структуры параметров:

Функция СведенияОВнешнейОбработке() Экспорт

ПараметрыРегистрации = Новый Структура;

ПараметрыРегистрации.Вставить("Вид", ...);

ПараметрыРегистрации.Вставить("Назначение", ...);

ПараметрыРегистрации.Вставить("Наименование", ...);

ПараметрыРегистрации.Вставить("Версия", ...);

ПараметрыРегистрации.Вставить("Информация", ...);

ПараметрыРегистрации.Вставить("ВерсияБСП", "1.2.1.4");

Команды = Новый ТаблицаЗначений;

Команды.Колонки.Добавить("Представление", Новый ОписаниеТипов("Строка"));

Команды.Колонки.Добавить("Идентификатор", Новый ОписаниеТипов("Строка"));

Команды.Колонки.Добавить("Использование", Новый ОписаниеТипов("Строка"));

Команды.Колонки.Добавить("ПоказыватьОповещение", Новый ОписаниеТипов("Булево"));

Команды.Колонки.Добавить("Модификатор", Новый ОписаниеТипов("Строка"));

ПараметрыРегистрации.Вставить("Команды", Команды);

Команда = ПараметрыРегистрации.Команды.Добавить();

Команда.Представление = НСтр("ru = '<Представление команды>'");

Команда.Идентификатор = "<Имя команды>";

Команда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКоманды<...>();

Команда.ПоказыватьОповещение = <Истина/Ложь>;

Возврат ПараметрыРегистрации;

КонецФункции

3. Для дополнительных отчетов принять решение по поводу подключения к подсистеме «Варианты отчетов» (если подсистема внедрена в конфигурацию).

Подробнее про возможности подсистемы «Варианты отчетов» см. в [соответствующем разделе](https://its.1c.ru/db/content/bsp301doc/src/3.7.%20валюты.htm?_=1542292879#_варианты_отчетов). В частности, варианты дополнительного отчета, подключенного к подсистеме «Варианты отчетов», можно размещать в панелях отчетов программы, рядом с отчетами конфигурации.

Для подключения отчета к подсистеме «Варианты отчетов» в свойстве Хранилище вариантов (меню Действия, команда Свойства) следует выбрать ХранилищеВариантовОтчетов.

Если дополнительный отчет подключен к подсистеме «Варианты отчетов» и использует общие формы ФормаОтчета, ФормаНастроекОтчета и ФормаВариантаОтчета, то в функции СведенияОВнешнейОбработке появляется возможность подключить обработчики, расширяющие поведение формы отчета. Для этого в свойстве ПараметрыРегистрации.ОпределитьНастройкиФормы следует указать значение Истина и в модуле объекта отчета 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

**Вариант запуска «Открытие формы»**

Для этого варианта запуска необходимо создать форму обработки, которая будет открыта при выполнении команды:

* ● для глобальных отчетов и обработок – не требуется дополнительных действий;
* ● для назначаемых отчетов и обработок – в форме обработки добавить параметр ОбъектыНазначения типа Произвольный. В этот параметр передается массив ссылок на объекты, для которых выполняется дополнительная обработка.

Примечание

При использовании варианта запуска Открытие формы требуется самостоятельно вызвать метод ОповеститьОбИзменении после изменения объекта(ов), для того чтобы отразить изменения в пользовательском интерфейсе.

В форму передаются два параметра: ИдентификаторКоманды и ДополнительнаяОбработкаСсылка (ссылка на элемент справочника ДополнительныеОтчетыИОбработки, который связан с данной дополнительной обработкой).

**Вариант запуска «Вызов клиентского метода»**

Для этого варианта запуска необходимо создать форму обработки и в модуле формы обработки завести экспортную процедуру определенного вида.

Для глобальных отчетов и обработок – реализовать экспортную процедуру ВыполнитьКоманду с параметром ИдентификаторКоманды (Строка):

&НаКлиенте

Процедура ВыполнитьКоманду(ИдентификаторКоманды)

  // Реализация логики команды

  // ...

КонецПроцедуры

Для назначаемых обработок типа ПечатнаяФорма реализовать экспортную процедуру Печать с двумя параметрами: ИдентификаторКоманды и ОбъектыНазначенияМассив, где ИдентификаторКоманды – строка, идентификатор команды; ОбъектыНазначенияМассив – массив ссылок на объекты информационной базы, для которых выполняется дополнительная обработка:

&НаКлиенте

Процедура Печать(ИдентификаторКоманды, ОбъектыНазначенияМассив)

  // Реализация логики команды печати

  // ...

КонецПроцедуры

Для назначаемых обработок типа Создание связанных объектов – реализовать экспортную процедуру ВыполнитьКоманду с параметрами ИдентификаторКоманды, ОбъектыНазначенияМассив и СозданныеОбъекты, где СозданныеОбъекты – массив ссылок на созданные объекты:

&НаКлиенте

Процедура ВыполнитьКоманду(ИдентификаторКоманды, ОбъектыНазначенияМассив, СозданныеОбъекты)

  // Реализация логики команды по созданию связанных объектов

  // ...

КонецПроцедуры

Для назначаемых обработок типа Заполнение объекта и Отчет реализовать экспортную процедуру ВыполнитьКоманду с параметрами ИдентификаторКоманды, ОбъектыНазначенияМассив:

&НаКлиенте

Процедура ВыполнитьКоманду(ИдентификаторКоманды, ОбъектыНазначенияМассив)

  // Реализация логики команды по заполнению объекта

  // ...

КонецПроцедуры

Для всех типов дополнительных отчетов и обработок в форму передается параметр ДополнительнаяОбработкаСсылка – ссылка на элемент справочника ДополнительныеОтчетыИОбработки, который связан с данной дополнительной обработкой или отчетом.

**Вариант запуска «Вызов серверного метода»**

Для этого варианта запуска необходимо в модуле обработки требуется завести экспортную процедуру определенного вида.

Для глобальных отчетов и глобальных обработок реализовать экспортную процедуру ВыполнитьКоманду с параметрами ИдентификаторКоманды и ПараметрыВыполненияКоманд (подробнее про состав параметров см. комментарий к функции ТипКомандыВызовСерверногоМетода модуля ДополнительныеОтчетыИОбработкиКлиентСервер):

Процедура ВыполнитьКоманду(ИдентификаторКоманды, ПараметрыВыполненияКоманды) Экспорт

  // Реализация логики команды

  Если ИдентификаторКоманды = ... Тогда

    ...

  ИначеЕсли ...

КонецПроцедуры

Для назначаемых обработок типа Создание связанных объектов реализовать экспортную процедуру ВыполнитьКоманду с параметрами ИдентификаторКоманды, ОбъектыНазначения, СозданныеОбъекты и ПараметрыВыполненияКоманды:

Процедура ВыполнитьКоманду(ИдентификаторКоманды, ОбъектыНазначения, СозданныеОбъекты, ПараметрыВыполненияКоманды) Экспорт

  // Реализация логики команды по созданию связанных объектов

  Если ИдентификаторКоманды = ... Тогда

    ...

  ИначеЕсли ...

КонецПроцедуры

Для назначаемых обработок типа ПечатнаяФорма реализовать экспортную процедуру Печать с параметрами МассивОбъектов, КоллекцияПечатныхФорм, ОбъектыПечати и ПараметрыВывода. Описание параметров см. в разделе «[Печать](https://its.1c.ru/db/content/bsp301doc/src/3.38.%20печать.htm?_=1542292879#_печать)». При этом в структуре ПараметрыВывода содержится свойство ДополнительнаяОбработкаСсылка (ссылка на элемент справочника ДополнительныеОтчетыИОбработки, который связан с данной дополнительной обработкой).

Процедура Печать(МассивОбъектов, КоллекцияПечатныхФорм, ОбъектыПечати, ПараметрыВывода) Экспорт

  // Реализация логики команды печати

  Если ИдентификаторКоманды = ... Тогда

    ...

  ИначеЕсли ...

КонецПроцедуры

**Пример: внешняя печатная форма**

Обработка с одной печатной формой:

Функция СведенияОВнешнейОбработке() Экспорт

ПараметрыРегистрации = ДополнительныеОтчетыИОбработки.СведенияОВнешнейОбработке("2.3.1.73");

ПараметрыРегистрации.Вид = ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработкиПечатнаяФорма();

ПараметрыРегистрации.Версия = "1.3";

// Определение объектов, к которым подключается эта обработка.

ПараметрыРегистрации.Назначение.Добавить("Документ.\_ДемоСчетНаОплатуПокупателю");

// Добавление команды печати "Счет на оплату".

НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();

НоваяКоманда.Представление = НСтр("ru = 'Счет на оплату (внешняя печатная форма)'");

НоваяКоманда.Идентификатор = "СчетНаОплату";

НоваяКоманда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКомандыВызовСерверногоМетода();

НоваяКоманда.Модификатор = "ПечатьMXL";

Возврат ПараметрыРегистрации;

КонецФункции

Процедура Печать(МассивОбъектов, КоллекцияПечатныхФорм, ОбъектыПечати, ПараметрыВывода) Экспорт

ПечатнаяФорма = УправлениеПечатью.СведенияОПечатнойФорме(КоллекцияПечатныхФорм, "СчетНаОплату");

Если ПечатнаяФорма <> Неопределено Тогда

ПечатнаяФорма.ТабличныйДокумент = СформироватьСчетНаОплатуПокупателю(МассивОбъектов, ОбъектыПечати);

ПечатнаяФорма.СинонимМакета = НСтр("ru = 'Счет на оплату'");

КонецЕсли;

КонецПроцедуры

Функция СформироватьСчетНаОплатуПокупателю(МассивОбъектов, ОбъектыПечати)

ТабличныйДокумент = Новый ТабличныйДокумент;

 ....

Возврат ТабличныйДокумент;

КонецФункции

Для подмены встроенной команды печати необходимо указать ее идентификатор в параметре ЗаменяемыеКоманды:

// Добавление команды печати "Счет на оплату".

НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();

...

НоваяКоманда.ЗаменяемыеКоманды = "СчетЗаказ";

Список имеющихся команд печати с идентификаторами находится, как правило, в процедуре ДобавитьКомандыПечати модуля менеджера объекта. Если подмену делать не надо, параметр можно не указывать.

Обработка может содержать несколько печатных форм:

Функция СведенияОВнешнейОбработке() Экспорт

ПараметрыРегистрации = ДополнительныеОтчетыИОбработки.СведенияОВнешнейОбработке("2.3.1.73");

ПараметрыРегистрации.Вид = ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработкиПечатнаяФорма();

ПараметрыРегистрации.Версия = "1.3";

// Определение объектов, к которым подключается эта обработка.

ПараметрыРегистрации.Назначение.Добавить("Документ.\_ДемоСчетНаОплатуПокупателю");

// Добавление команды печати "Счет на оплату".

НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();

НоваяКоманда.Представление = НСтр("ru = 'Счет на оплату (внешняя печатная форма)'");

НоваяКоманда.Идентификатор = "СчетНаОплату";

НоваяКоманда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКомандыВызовСерверногоМетода();

НоваяКоманда.Модификатор = "ПечатьMXL";

НоваяКоманда.ЗаменяемыеКоманды = "СчетЗаказ";

// Добавление команды печати "Заказ покупателя".

НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();

НоваяКоманда.Представление = НСтр("ru = 'Заказ покупателя (внешняя печатная форма)'");

НоваяКоманда.Идентификатор = "ЗаказПокупателя";

НоваяКоманда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКомандыВызовСерверногоМетода();

НоваяКоманда.Модификатор = "ПечатьMXL";

Возврат ПараметрыРегистрации;

КонецФункции

Процедура Печать(МассивОбъектов, КоллекцияПечатныхФорм, ОбъектыПечати, ПараметрыВывода) Экспорт

ПечатнаяФорма = УправлениеПечатью.СведенияОПечатнойФорме(КоллекцияПечатныхФорм, "СчетНаОплату");

Если ПечатнаяФорма <> Неопределено Тогда

ПечатнаяФорма.ТабличныйДокумент = СформироватьСчетНаОплатуПокупателю(МассивОбъектов, ОбъектыПечати);

ПечатнаяФорма.СинонимМакета = НСтр("ru = 'Счет на оплату'");

КонецЕсли;

ПечатнаяФорма = УправлениеПечатью.СведенияОПечатнойФорме(КоллекцияПечатныхФорм, "ЗаказПокупателя");

Если ПечатнаяФорма <> Неопределено Тогда

ПечатнаяФорма.ТабличныйДокумент = СформироватьЗаказПокупателя(МассивОбъектов, ОбъектыПечати);

ПечатнаяФорма.СинонимМакета = НСтр("ru = 'Заказ покупателя (внешняя печатная форма)'");

КонецЕсли;

КонецПроцедуры

Функция СформироватьСчетНаОплатуПокупателю(МассивОбъектов, ОбъектыПечати)

ТабличныйДокумент = Новый ТабличныйДокумент;

....

Возврат ТабличныйДокумент;

КонецФункции

Функция СформироватьЗаказПокупателя(МассивОбъектов, ОбъектыПечати)

ТабличныйДокумент = Новый ТабличныйДокумент;

....

Возврат ТабличныйДокумент;

КонецФункции

Обработка может содержать команду печати комплекта печатных форм. Описание команды печати комплекта отличается от команды печати одной печатной формы тем, что параметр Идентификатор содержит список идентификаторов печатных форм (этой обработки), входящих в комплект:

// Добавление команды печати "Комплект документов".

НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();

НоваяКоманда.Представление = НСтр("ru = 'Комплект документов (внешняя печатная форма)'");

НоваяКоманда.Идентификатор = "СчетНаОплату,ЗаказПокупателя,ГарантийноеПисьмо";

НоваяКоманда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКомандыВызовСерверногоМетода();

НоваяКоманда.Модификатор = "ПечатьMXL";

Пример обработки с командой печати комплекта:

Функция СведенияОВнешнейОбработке() Экспорт

ПараметрыРегистрации = ДополнительныеОтчетыИОбработки.СведенияОВнешнейОбработке("2.3.1.73");

ПараметрыРегистрации.Вид = ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработкиПечатнаяФорма();

ПараметрыРегистрации.Версия = "1.3";

// Определение объектов, к которым подключается эта обработка.

ПараметрыРегистрации.Назначение.Добавить("Документ.\_ДемоСчетНаОплатуПокупателю");

// Добавление команды печати "Комплект документов".

НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();

НоваяКоманда.Представление = НСтр("ru = 'Комплект документов (внешняя печатная форма)'");

НоваяКоманда.Идентификатор = "СчетНаОплату,ЗаказПокупателя,ГарантийноеПисьмо";

НоваяКоманда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКомандыВызовСерверногоМетода();

НоваяКоманда.Модификатор = "ПечатьMXL";

Возврат ПараметрыРегистрации;

КонецФункции

Процедура Печать(МассивОбъектов, КоллекцияПечатныхФорм, ОбъектыПечати, ПараметрыВывода) Экспорт

ПечатнаяФорма = УправлениеПечатью.СведенияОПечатнойФорме(КоллекцияПечатныхФорм, "СчетНаОплату");

Если ПечатнаяФорма <> Неопределено Тогда

ПечатнаяФорма.ТабличныйДокумент = СформироватьСчетНаОплатуПокупателю(МассивОбъектов, ОбъектыПечати);

ПечатнаяФорма.СинонимМакета = НСтр("ru = 'Счет на оплату'");

КонецЕсли;

ПечатнаяФорма = УправлениеПечатью.СведенияОПечатнойФорме(КоллекцияПечатныхФорм, "ЗаказПокупателя");

Если ПечатнаяФорма <> Неопределено Тогда

ПечатнаяФорма.ТабличныйДокумент = СформироватьЗаказПокупателя(МассивОбъектов, ОбъектыПечати);

ПечатнаяФорма.СинонимМакета = НСтр("ru = 'Заказ покупателя (внешняя печатная форма)'");

КонецЕсли;

ПечатнаяФорма = УправлениеПечатью.СведенияОПечатнойФорме(КоллекцияПечатныхФорм, "ГарантийноеПисьмо");

Если ПечатнаяФорма <> Неопределено Тогда

ПечатнаяФорма.ТабличныйДокумент = СформироватьГарантийноеПисьмо(МассивОбъектов, ОбъектыПечати);

ПечатнаяФорма.СинонимМакета = НСтр("ru = 'Гарантийное письмо (внешняя печатная форма)'");

КонецЕсли;

КонецПроцедуры

Функция СформироватьСчетНаОплатуПокупателю(МассивОбъектов, ОбъектыПечати)

ТабличныйДокумент = Новый ТабличныйДокумент;

....

Возврат ТабличныйДокумент;

КонецФункции

Функция СформироватьЗаказПокупателя(МассивОбъектов, ОбъектыПечати)

ТабличныйДокумент = Новый ТабличныйДокумент;

....

Возврат ТабличныйДокумент;

КонецФункции

Функция СформироватьГарантийноеПисьмо(МассивОбъектов, ОбъектыПечати)

ТабличныйДокумент = Новый ТабличныйДокумент;

....

Возврат ТабличныйДокумент;

КонецФункции

Для назначаемых обработок типа Заполнение объекта и других типов обработок необходимо реализовать экспортную процедуру ВыполнитьКоманду с параметрами ИдентификаторКоманды, ОбъектыНазначения и ПараметрыВыполненияКоманды:

Процедура ВыполнитьКоманду(ИдентификаторКоманды, ОбъектыНазначения, ПараметрыВыполненияКоманды) Экспорт

  // Реализация логики команды по заполнению объекта

  Если ИдентификаторКоманды = ... Тогда

    ...

  ИначеЕсли ...

КонецПроцедуры

**Пример: обработка заполнения данных формы без записи объекта**

Для типа обработок ЗаполнениеОбъекта предусмотрена возможность работать напрямую с данными формы (а не со ссылкой на объект), что позволяет не выполнять принудительную запись объекта в форме ни до, ни после выполнения обработки.

Пример команды заполнения данных формы:

Функция СведенияОВнешнейОбработке() Экспорт

ПараметрыРегистрации =ДополнительныеОтчетыИОбработки.СведенияОВнешнейОбработке("2.2.2.1");

ПараметрыРегистрации.Вид =ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработкиЗаполнениеОбъекта();

ПараметрыРегистрации.Версия = "1.2";

ПараметрыРегистрации.Назначение.Добавить("Справочник.\_ДемоКонтрагенты");

НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();

НоваяКоманда.Представление = НСтр("ru = 'Заполнить реквизит ""ИНН"" не записывая объект (заполнение формы)'");

НоваяКоманда.Идентификатор = "ЗаполнитьИНН";

НоваяКоманда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКомандыЗаполнениеФормы();

Возврат ПараметрыРегистрации;

КонецФункции

Процедура ВыполнитьКоманду(ИмяКоманды, ОбъектыНазначения, ПараметрыВыполнения) Экспорт

Если ИмяКоманды = "ЗаполнитьИНН" Тогда

Генератор = Новый ГенераторСлучайныхЧисел;

ЭтаФорма = ПараметрыВыполнения.ЭтаФорма;

ЭтаФорма.Объект.ИНН = Формат(Генератор.СлучайноеЧисло(1, 999999999), "ЧЦ=12; ЧДЦ=0; ЧВН=; ЧГ=");

ЭтаФорма.Модифицированность = Истина;

Сообщение = Новый СообщениеПользователю();

Сообщение.Поле = "Объект.ИНН";

Сообщение.Текст = НСтр("ru = 'Поле ""ИНН"" успешно заполнено'");

Сообщение.Сообщить();

...

КонецПроцедуры

Подробнее см. дополнительную обработку Демо: Обработка заполнения в справочнике Дополнительные отчеты и обработки демонстрационной конфигурации.

**Пример: сохраняемые параметры дополнительной обработки**

При необходимости задать параметры для дополнительной обработки можно воспользоваться реквизитом ХранилищеНастроек справочника ДополнительныеОтчетыИОбработки. Например, для заполнения документа Распределение прочих затрат необходимо получать цены по определенному, специально созданному для этих целей виду цен. Этот вид цен можно задавать с помощью сохраняемого параметра. В общем виде в реквизите ХранилищеНастроек можно сохранять произвольные параметры дополнительной обработки .(например, в виде структуры). Для доступа к реквизиту ХранилищеНастроек в команду выполнения обработки передается ссылка на связанный с ней элемент справочника ДополнительныеОтчетыИОбработки.

Пример кода для чтения параметра:

ХранилищеНастроек = ОбщегоНазначения.ЗначениеРеквизита(ОбработкаСсылка,"ХранилищеНастроек");

НастройкиСтруктура = ХранилищеНастроек.Получить();

для записи:

ДополнительнаяОбработкаОбъект = Параметры.ДополнительнаяОбработкаСсылка.ПолучитьОбъект();

ДополнительнаяОбработкаОбъект.ХранилищеНастроек = НовыйХранилищеЗначения(НастройкиСтруктура);

ДополнительнаяОбработкаОбъект.Записать();

Подробнее см. дополнительные обработки Демо: Загрузка номенклатуры из прайс-листа в справочнике Дополнительные отчеты и обработки демонстрационной конфигурации.

**Фоновое выполнение длительных операций**

Для того чтобы длительные операции дополнительного объекта запускались в фоновом режиме, необходимо запустить выполнение команды в фоновом задании и после его завершения принять результат.

Программный интерфейс, помогающий запускать длительные операции в фоне, представлен следующими процедурами:

1. В общем модуле ДополнительныеОтчетыИОбработкиКлиент: процедуры ВыполнитьКомандуВФоне и ПараметрыВыполненияКомандыВФоне.

2. В общем модуле ДополнительныеОтчетыИОбработки: ВыполнитьКомандуИзФормыВнешнегоОбъекта и ВыполнитьКоманду.

Рассмотрим подключение по шагам.

Шаг 1. В форме дополнительной обработки добавить ключевые параметры:

- "ДополнительнаяОбработкаСсылка" с типом "СправочникСсылка.ДополнительныеОтчетыИОбработки" для хранения ссылки внешнего объекта.

- "ИдентификаторКоманды" с типом "Строка" для хранения идентификатора выполняемой команды.

Шаг 2. Запустить длительную операцию при помощи кода:

ПараметрыКоманды = ДополнительныеОтчетыИОбработкиКлиент.ПараметрыВыполненияКомандыВФоне(Параметры.ДополнительнаяОбработкаСсылка);

ПараметрыКоманды.СопровождающийТекст = НСтр("ru = '<Вставить представление выполняемой команды...>'");

Обработчик = Новый ОписаниеОповещения("ПослеЗавершенияДлительнойОперации", ЭтотОбъект, СопровождающийТекст);

ДополнительныеОтчетыИОбработкиКлиент.ВыполнитьКомандуВФоне(Параметры.ИдентификаторКоманды, ПараметрыКоманды, Обработчик);

Шаг 3. Добавить процедуру для приемки результата:

&НаКлиенте

Процедура ПослеЗавершенияДлительнойОперации(Операция, СопровождающийТекст) Экспорт

// Добавить код приемки результата.

// Например:

//Если ТипЗнч(ВладелецФормы) = Тип("УправляемаяФорма") И Не ВладелецФормы.Модифицированность Тогда

// Попытка

// ВладелецФормы.Прочитать();

// Исключение

// // Значит это форма списка.

// КонецПопытки;

//КонецЕсли;

//Если Открыта() Тогда

// Закрыть();

//КонецЕсли;

//Если Операция.Статус = "Выполнено" Тогда

// ПоказатьОповещениеПользователя(НСтр("ru = 'Успешное завершение'"), , СопровождающийТекст, БиблиотекаКартинок.Успешно32);

//Иначе

// Предупреждение = Новый Структура;

// Предупреждение.Вставить("Текст",    Операция.КраткоеПредставлениеОшибки);

// Предупреждение.Вставить("Подробно", Операция.ПодробноеПредставлениеОшибки);

// СтандартныеПодсистемыКлиент.ВывестиПредупреждение(ВладелецФормы, Предупреждение);

//КонецЕсли;

КонецПроцедуры

Шаг 4. В модуле объекта добавить обработчик команды "Вызов серверного метода" по [шаблону](https://its.1c.ru/db/content/bsp301doc/src/3.16.%20%D0%B4%D0%BE%D0%BF%D0%BE%D0%BB%D0%BD%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D1%8B%20%D0%B8%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8.htm?_=1542292879#_вариант_запуска_«вызов).

Примечание

Длительные операции могут быть только серверными.

В демонстрационной базе БСП данная возможность представлена в обработках Демо: Управление полнотекстовым поиском, Демо: Загрузка номенклатуры из прайс-листа (профили безопасности), Демо: Обработка заполнения и Демо: Создание связанных объектов в справочнике Дополнительные отчеты и обработки.

**Настройка обмена данными**

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включать в планы обмена распределенной ИБ.

Для планов обмена автономного рабочего места (АРМ) в модели сервиса константу ИспользоватьДополнительныеОтчетыИОбработки следует включать в состав планов обмена, но не включать в состав подписок на события (выполняющих регистрацию данных при записи). Данное требование объясняется тем, что в автономном рабочем месте использование дополнительных отчетов и обработок настраивается независимо от их использования в экземпляре сервиса и должно передаваться только при создании начального образа.