**3.26. Контроль ведения учета**

Подсистема «Контроль ведения учета» предназначена для автоматического регламентного контроля корректности данных информационной базы по произвольным прикладным правилам, которые дополняют сущестувующие проверки при записи документов и других объектов конфигурации. Например: нарушение ссылочной целостности, отрицательные остатки в регистре накопления, сбой в нумерации счетов фактур и т.д., а также вывода выявленных проблем для различных категорий пользователей. Например, если это операция закрытия месяца, то проверку и исправление ошибок должен проводить пользователь, компетентный в вопросах закрытия месяца, если это проверка на наличие «битых» ссылок или тестирование/исправление, то такую проверку должен проводить администратор системы и т.д. Контроль выполняется в фоне, по расписанию, или может быть явно инициирован пользователем.

**Настройка**

Принять решение по поводу состава прикладных проверок, которые дополнительно требуется выполнять для контроля ведения учета, помимо существующих проверок заполнения при записи объектов. Например, это могут быть сложные проверки, которые контролируют состояние нескольких связанных объектов и которые целесообразно выполнять в фоне (а не во время работы пользователя), или которые следует выполнять только на определенных этапах – при операции закрытия месяца, перед отправкой отчетности и т.п. По умолчанию, в подсистеме поставляются системные проверки, которые автоматически контролируют  корректность данных в фоне:

● Поиск ссылок на несуществующие файлы в томах хранения.

● Проверка дублирования предопределенных элементов.

● Проверка незаполненных обязательных реквизитов.

● Проверка отсутствия предопределенных узлов плана обмена.

● Проверка отсутствующих предопределенных элементов.

● Проверка ссылочной целостности.

● Проверка циклических ссылок.

Для добавления прикладной проверки необходимо:

● В процедуру ПриОпределенииПроверок общего модуля КонтрольВеденияУчетаПереопределяемый внести описание проверки в следующем виде:

Проверка = Проверки.Добавить();

Проверка.ИдентификаторГруппы          = "СистемныеПроверки";

Проверка.Наименование                 = НСтр("ru='Демо: Проверка заполнения комментария в документах ""Демо: Поступление товаров""'");

Проверка.Причины                      = НСтр("ru='Не введен комментарий в документе.'");

Проверка.Рекомендация                 = НСтр("ru='Ввести комментарий в документе.'");

Проверка.Идентификатор                = "ПроверитьКомментарийВПоступленииТоваров";

Проверка.ОбработчикПроверки           = "\_ДемоСтандартныеПодсистемы.ПроверитьКомментарийВПоступленииТоваров";

Проверка.ДатаНачалаПроверки           = Дата('20140101000000');

Проверка.ЛимитПроблем                 = 3;

Проверка.ЗапрещеноИзменениеВажности   = Истина;

Проверка.КонтекстПроверокВеденияУчета = "СистемныеПроверки";

● В свойстве ОбработчикПроверки указать полный путь экспортной процедуры-обработчика проверки в формате <СерверныйОбщийМодуль>.<ИмяПроцедуры>. Пример реализации см. в \_ДемоСтандартныеПодсистемы. ПроверитьКомментарийВПоступленииТоваров.

● Подробное описание свойств проверки находятся в комментарии к процедуре ПриОпределенииПроверок общего модуля КонтрольВеденияУчетаПереопределяемый;

● В общем случае формат обработчика проверки должен иметь следующий вид:

Процедура ПроверитьКомментарийВПоступленииТоваров(Проверка, ПараметрыПроверки) Экспорт

Результат = ОсуществитьПроверку();

Пока Результат.Следующий() Цикл

Проблема = МодульКонтрольВеденияУчета.ОписаниеПроблемы(Результат.Ссылка, ПараметрыПроверки);

Проблема.Ответственный = Результат.Ответственный;

Проблема.УточнениеПроблемы = ?(ЗначениеЗаполнено(Результат.Комментарий),

НСтр("ru = 'В комментарии введены пробелы или табуляции.'"),

НСтр("ru = 'Не введен комментарий в документе.'"));

МодульКонтрольВеденияУчета.ЗаписатьПроблему(Проблема, ПараметрыПроверки);

КонецЦикла;

КонецПроцедуры

● Вместе с проверкой подсистема также позволяет реализовать алгоритм по исправлению выявленных ее проблем. Пользователи могут переходить к исправлению из отчета РезультатыПроверкиУчета.

● Для открытия формы (рабочего места), в которой предусмотрена процедура исправления в описании проверки, в свойстве ОбработчикПереходаКИсправлению указать:

Проверка.ОбработчикПереходаКИсправлению = "Отчет.РезультатыПроверкиУчета.Форма.ИсправлениеЦиклическихСсылок";

● При этом в форму будут переданы параметры ИдентификаторПроверки и ВидПроверки, по которым в коде модуля формы можно определить, исправление каких именно проблем запросил пользователь;

● Либо в свойстве ОбработчикПереходаКИсправлению можно указать экспортную процедуру-обработчик клиентского общего модуля:

Проверка.ОбработчикПереходаКИсправлению = " КонтрольВеденияУчетаСлужебныйКлиент.ИсправлениеЦиклическихСсылок";

● При этом в экспортной клиенстской процедуре-обработчике должны быть определены два параметра ИдентификаторПроверки и ВидПроверки:

Продедура ИсправлениеЦиклическихСсылок(ИдентификаторПроверки, ВидПроверки) Экспорт

Затем необходимо определить ссылочные объекты метаданных конфигурации (справочники, документы и т.п.), в формах и списках которых требуется выводить информацию о выявленных в них проблемах. Например, это могут быть вообще все объекты конфигурации, либо только те, которые наиболее критичны для целей контроля ведения учета. Для каждого такого объекта необходимо:

● В модуле формы списка в процедуру ПриСозданииНаСервере добавить фрагмент:

// СтандартныеПодсистемы.КонтрольВеденияУчета

КонтрольВеденияУчета.ПриСозданииНаСервереФормыСписка(ЭтотОбъект, "Список");

// Конец СтандартныеПодсистемы.КонтрольВеденияУчета

● У динамического списка добавить обработчик события СписокПриПолученииДанныхНаСервере:

// СтандартныеПодсистемы.КонтрольВеденияУчета

КонтрольВеденияУчета.ПриПолученииДанныхНаСервере(Настройки, Строки);

// Конец СтандартныеПодсистемы.КонтрольВеденияУчета

● И в область СлужебныеПроцедурыИФункции модуля формы списка поместить следующий код:

// СтандартныеПодсистемы.КонтрольВеденияУчета

&НаКлиенте

Процедура Подключаемый\_Выбор(Элемент, ВыбраннаяСтрока, Поле, СтандартнаяОбработка) Экспорт

КонтрольВеденияУчетаКлиент.ОткрытьОтчетПоПроблемамИзСписка(ЭтотОбъект, "Список", Поле, СтандартнаяОбработка);

КонецПроцедуры

// Конец СтандартныеПодсистемы.КонтрольВеденияУчета

● В модуле формы объекта в процедуру ПриЧтенииНаСервере необходимо добавить фрагмент:

// СтандартныеПодсистемы.КонтрольВеденияУчета

КонтрольВеденияУчета.ПриЧтенииНаСервере(ЭтотОбъект, ТекущийОбъект);

// Конец СтандартныеПодсистемы.КонтрольВеденияУчета

● В область СлужебныеПроцедурыИФункции модуля формы объекта поместить следующий код:

// СтандартныеПодсистемы.КонтрольВеденияУчета

&НаКлиенте

Процедура Подключаемый\_ОткрытьОтчетПоПроблемам(ЭлементИлиКоманда, НавигационнаяСсылка, СтандартнаяОбработка)

КонтрольВеденияУчетаКлиент.ОткрытьОтчетПоПроблемамОбъекта(ЭтотОбъект, Объект.Ссылка, СтандартнаяОбработка);

КонецПроцедуры

// Конец СтандартныеПодсистемы.КонтрольВеденияУчета

По умолчанию, информация о выявленных проблемах выводится в списках в виде колонки с восклицательным знаком, а также в верхней части форм самих объектов. Однако это можно переопределить с помощью общего модуля КонтрольВеденияУчетаПереопределяемый.

Если в конфигурации не используется подсистема «Настройки программы», то в рабочем месте администратора программы необходимо разместить команду открытия справочника ПравилаПроверкиУчета.

Если в конфигурации не используется подсистема «Текущие дела», то рекомендуется уведомлять пользователей о выявленных проблемах на начальной странице другими средствами, воспользовавшись программным интерфейсом ОткрытьОтчетПоПроблемам, ОткрытьОтчетПоПроблемамОбъекта или ОткрытьОтчетПоПроблемамИзСписка общего модуля КонтрольВеденияУчетаКлиент.

**Особые случаи внедрения**

При внедрении подсистемы может возникнуть необходимость ограничить доступ на уровне записей к сведениям о выявленных проблемах. Например, если в конфигурации предусмотрена настройка доступа к данным в разрезе организаций и проектов, то и доступ к сведениям о выявленных проблемах также должен быть ограничен соответственным образом. Для этого необходимо:

● добавить в регистр РезультатыПроверкиУчета реквизиты, по которым будет осуществляться доступ, и вписать в роль ЧтениеРезультатовПроверкиУчета в право Чтение требуемые ограничения доступа к данным;

● заполнять эти реквизиты при записи проблем из кода прикладных проверок. Например, переносить их непосредственно из проблемного объекта, или по более сложным правилам. При этом для поставляемых проверок следует воспользоваться процедурой ПередЗаписьюПроблемы общего модуля КонтрольВеденияУчетаПереопределяемый, в которой разместить код по заполнению следующего вида:

Если Реквизиты.Найти("Организация") <> Неопределено Тогда

Проблема.Вставить("\_ДемоОрганизация", ОбщегоНазначения.ЗначениеРеквизитаОбъекта(СсылкаНаОбъект, "Организация"));

КонецЕсли;При использовании подсистемы может возникнуть необходимость группировать проверки по какому-либо прикладному признаку. Например, нужно выделить все проверки, которые относятся к оперции расчета заработной платы. Для этого необходимо добавить в определяемый тип КонтекстПроверокВеденияУчета интересующий тип метаданных, после чего в процедуре ПриОпределенииПроверок общего модуля КонтрольВеденияУчетаПереопределяемый присвоить свойству КонтекстПроверокВеденияУчета  нужное значение.

Кроме того, при необходимости этот признак (контекст проверки) может быть дополнительно уточнен еще одним значением. Например, уточнение может быть в виде вопроса «Когда выполнять проверку?»  с возможными вариантами ответа: «До расчета заработной платы», «Во время расчета заработной платы» или «После расчета заработной платы». При необходимости такого уточнения нужно добавить в определяемый тип УточнениеКонтекстаПроверокВеденияУчета интересующий тип метаданных и задать одноименное свойство проверки по аналогии со свойством КонтекстПроверокВеденияУчета.

**Настройка прав доступа пользователей**

Для настройки прав доступа пользователей к данным подсистемы «Контроль ведения учета» следует использовать роли, приведенные ниже.

Таблица 3.51.

| № | Роли и их назначение |
| --- | --- |
| 1. | ПолныеПрава (из подсистемы «Базовая функциональность»)  Включение и отключение использования подсистемы «Контроль ведения учета». Изменение настроек выполняемых проверок. |
| 2. | ЧтениеРезультатовПроверкиУчета  Просмотр выявленных проблем, работа с отчетом «Контроль ведения учета» |

**Использование при разработке конфигурации**

Программный интерфейс подсистемы представлен в общих модулях КонтрольВеденияУчета и КонтрольВеденияУчетаКлиент.

Для просмотра выявленных проблем ведения учета подсистема включает в себя отчет РезультатыПроверкиУчета. В тех случаях, когда список проблем необходимо вывести в собственном рабочем месте (например, в форме закрытия месяца выводить проблемы, препятствующие закрытию месяца), следует воспользоваться процедурой ВыполнитьПроверку для программного выполнения нужной проверки и функциями  СводнаяИнформацияПоВидамПроверок или ПодробнаяИнформацияПоВидамПроверок для получения результатов проверки. При необходимости выполнить группу проверок, связанных общим признаком (например, нужно выполнить все проверки, относящиеся к операции закрытия месяца), следует воспользоваться процедурой ВыполнитьПроверкиВКонтексте.

**Общие рекомендации по разработке проверок**

При разработке собственных проверок рекомендуется

● исключать из проверки объекты, недоступные в даный момент по функциональным опциям, с помощью функции ОбъектМетаданныхДоступенПоФункциональнымОпциям общего модуля ОбщегоНазначения. Пример см. в процедуре ПроверитьНезаполненныеОбязательныеРеквизиты общего модуля КонтрольВеденияУчетаСлужебный;

● при работе в модели сервиса следует исключать из проверки объекты, не доступные роли ПолныеПрава, а также неразделенные данные с помощью функции ЭтоРазделенныйОбъектМетаданных общего модуля РаботаВМоделиСервиса. Кроме того, ряд проверок вообще не имеет смысла выполнять в модели сервиса, например, проверка ссылочной целостности отключена в модели сервиса, так как такая проверка должна выполняться централизованно администратом сервиса, а не администраторами областей данных.

**Предварительная очистка проблем ведения учета перед запуском новой проверки**

Для разработки проверок ведения учета, регистрирующих проблемы сразу нескольких видов (например, проблемы по указанному контрагенту или проблемы, относящиеся к определенному периоду), в общем модуле КонтрольВеденияУчета предусмотрена процедура ОчиститьРезультатыПредыдущихПроверок. С ее помощью следует предварительно очищать ранее зарегистрированные проблемы ведения учета в начале основного алгоритма обработчика проверки. В противном случае, одна и та же проблема может быть зарегистрирована многократно при нескольких последовательных запусках проверки.

При этом для непараметрических проверок предварительная очистка результатов предыдущих запусков выполняется автоматически, поэтому в их обработчиках вызов процедуры ОчиститьРезультатыПредыдущихПроверок не требуется.

**Настройка обмена данными**

Все объекты метаданных подсистемы, содержащие данные, не рекомендуется включать в планы обмена распределенной ИБ (РИБ), так как в каждом узле проверка ведения учета ведется независимо.