**Базовые приемы работы с кластером 1С при помощи БСП**

Публикация № **1537450**

**В данной публикации я рассматриваю базовые приемы работы с кластером серверных баз 1С, используя типовые типовые возможности библиотеки стандартных подсистем (БСП).**

**Введение**

Всем доброго времени суток!

В данной публикации я предлагаю рассмотреть базовые возможности актуальной **библиотеки стандартных подсистем (БСП)** в области функционала "**Администрирование кластера**". На момент написания статьи актуальная версия БСП - **это БСП версии 3.1.5.208**.

Данная публикация будет полезна прежде всего программистам-разработчикам, работающим с любыми современными типовыми конфигурациями в серверном режиме на Платформе 1С 8.3 и придерживающимся стандартов разработки. Методы, описываемые в статье, будут одинаково работать в любой современной типовой конфигурации с БСП.

**Примеры данной статьи будут работать при условии, что у вас серверная база. Все примеры приведены для COM-соединения.**

Изучение типовых типовых методов библиотеки стандартных подсистем, значительно сокращая время разработки обработок и расширений, не заставляя вас изобретать что-то свое.

Данный материал я условно разделил на две части, в которых я привожу два полезных практических примера: часть 1 - это сеансы и их удаление по фильтру, а часть 2 - это установка блокировок сеансов и регламентных заданий:

**Часть 1. Получение сеансов информационной базы данных и их удаление по заданному фильтру.**

Для начала рассмотрим одну из основных функций

**СеансыИнформационнойБазы(Знач ПараметрыАдминистрированияКластера, Знач ПараметрыАдминистрированияИБ, Знач Фильтр = Неопределено)**

Данная функция возвращает структуру, содержащую сеансы подключения к информационной базе данных, например:

**СеансыИнформационнойБазы**

// конструктор соединения к кластеру

АдминКластера = АдминистрированиеКластера.ПараметрыАдминистрированияКластера();

АдминКластера.АдресАгентаСервера = "localhost";

АдминКластера.ИмяАдминистратораКластера = "admin";

АдминКластера.ПарольАдминистратораКластера= "123456";

// конструктор соединения к базе данных

АдминИБ = АдминистрированиеКластера

.ПараметрыАдминистрированияИнформационнойБазыКластера();

АдминИБ.ИмяВКластере = "ssl83test";

АдминИБ.ИмяАдминистратораИнформационнойБазы = "Администратор";

АдминИБ.ПарольАдминистратораИнформационнойБазы = "123456";

// получим сеансы соединения с ИБ

СеансыИБ = АдминистрированиеКластераCOM

.СеансыИнформационнойБазы(АдминКластера, АдминИБ);

В результате выполнения данной функции я получу массив сеансов, которые подключены к выбранной базе данных. Выглядит это вот так:

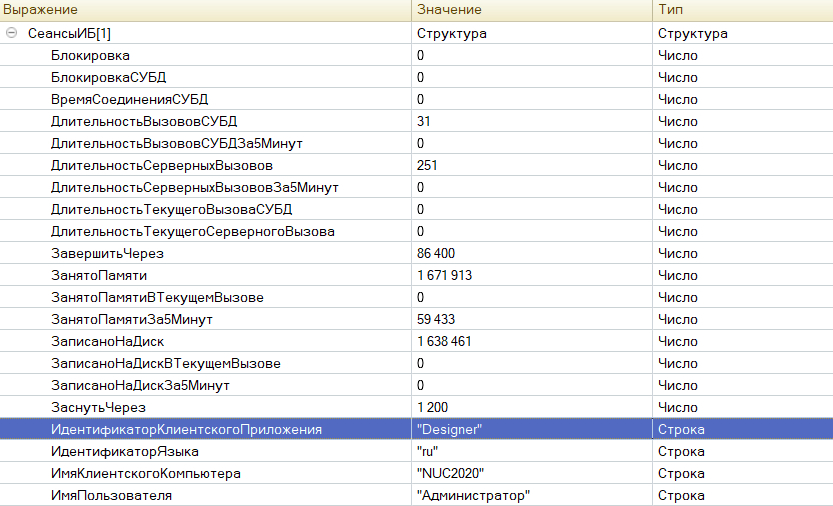


Рис.1 Структура сеанса, полученная функцией БСП СеансыИнформационнойБазы().

Теперь попробуем удалить полученные сеансы с помощью процедуры

**УдалитьСеансыИнформационнойБазы(**

**Знач ПараметрыАдминистрированияКластера,**

**Знач ПараметрыАдминистрированияИБ,**

**Знач Фильтр = Неопределено) Экспорт**

фильтр сеанса я буду создавать структурой:

**Фильтр сеансов информационной базы данных**

ФильтрУдаления = Новый Структура("ИдентификаторКлиентскогоПриложения");

ФильтрУдаления.ИдентификаторКлиентскогоПриложения = "Designer";

используем фильтр для удаления сеанса пользователя, который занял конфигуратор:

**Удаление сеанса "конфигуратора":**

АдминКластера = АдминистрированиеКластера.ПараметрыАдминистрированияКластера();

АдминКластера.АдресАгентаСервера = "localhost";

АдминКластера.ИмяАдминистратораКластера = "admin";

АдминКластера.ПарольАдминистратораКластера= "77777";

АдминИБ = АдминистрированиеКластера

.ПараметрыАдминистрированияИнформационнойБазыКластера();

АдминИБ.ИмяВКластере = "ssl83test";

АдминИБ.ИмяАдминистратораИнформационнойБазы = "Администратор";

АдминИБ.ПарольАдминистратораИнформационнойБазы = "123456";

// получим сеансы соединения с ИБ

СеансыИБ = АдминистрированиеКластераCOM

.СеансыИнформационнойБазы(АдминКластера, АдминИБ);

ФильтрУдаления = Новый Структура("ИдентификаторКлиентскогоПриложения");

ФильтрУдаления.ИдентификаторКлиентскогоПриложения = "Designer";

//выкидываем занявшего конфигуратор

АдминистрированиеКластераCOM.УдалитьСеансыИнформационнойБазы(АдминКластера, АдминИБ, ФильтрУдаления);

После применения данной процедуры у сотрудника, который занял конфигуратор и "ушел домой" выскочит вот такая известная надпись:

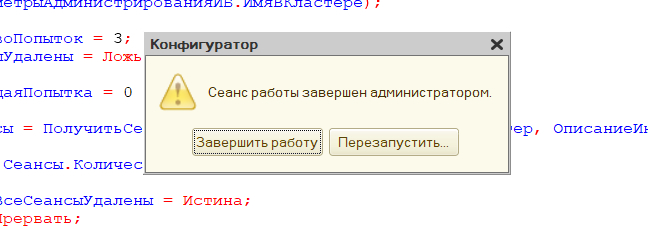


Рис.2.Выкидываем пользователя из конфигуратора.

т.е. не имея доступа к "серверу 1с" и ведя, с позволения сказать, "общую разработку без хранилища" в одной базе - можно выкинуть вот так, что называется "без вызванивания кого-то там".

Также возможна фильтрация и по другим параметрам. Вот некоторые из них:

**Словарь свойств сеансов**

Результат = АдминистрированиеКластера.СвойстваСеанса();

Результат.Номер = "SessionID";

Результат.ИмяПользователя = "UserName";

Результат.ИмяКлиентскогоКомпьютера = "Host";

Результат.ИдентификаторКлиентскогоПриложения = "AppID";

Результат.ИдентификаторЯзыка = "Locale";

Результат.МоментСозданияСеанса = "StartedAt";

Результат.МоментПоследнейАктивностиСеанса = "LastActiveAt";

Результат.БлокировкаСУБД = "blockedByDBMS";

Результат.Блокировка = "blockedByLS";

Результат.Передано = "bytesAll";

Результат.ПереданоЗа5Минут = "bytesLast5Min";

Результат.СерверныхВызовов = "callsAll";

Результат.СерверныхВызововЗа5Минут = "callsLast5Min";

Результат.ДлительностьСерверныхВызовов = "durationAll";

Результат.ДлительностьТекущегоСерверногоВызова = "durationCurrent";

Результат.ДлительностьСерверныхВызововЗа5Минут = "durationLast5Min";

Результат.ПереданоСУБД = "dbmsBytesAll";

Результат.ПереданоСУБДЗа5Минут = "dbmsBytesLast5Min";

Результат.ДлительностьВызововСУБД = "durationAllDBMS";

Результат.ДлительностьТекущегоВызоваСУБД = "durationCurrentDBMS";

Результат.ДлительностьВызововСУБДЗа5Минут = "durationLast5MinDBMS";

Результат.СоединениеСУБД = "dbProcInfo";

Результат.ВремяСоединенияСУБД = "dbProcTook";

Результат.МоментЗахватаСоединенияСУБД = "dbProcTookAt";

Результат.Спящий = "Hibernate";

Результат.ЗавершитьЧерез = "HibernateSessionTerminateTime";

Результат.ЗаснутьЧерез = "PassiveSessionHibernateTime";

Результат.ПрочитаноСДиска = "InBytesAll";

Результат.ПрочитаноСДискаВТекущемВызове = "InBytesCurrent";

Результат.ПрочитаноСДискаЗа5Минут = "InBytesLast5Min";

Результат.ЗанятоПамяти = "MemoryAll";

Результат.ЗанятоПамятиВТекущемВызове = "MemoryCurrent";

Результат.ЗанятоПамятиЗа5Минут = "MemoryLast5Min";

Результат.ЗаписаноНаДиск = "OutBytesAll";

Результат.ЗаписаноНаДискВТекущемВызове = "OutBytesCurrent";

Результат.ЗаписаноНаДискЗа5Минут = "OutBytesLast5Min";

Результат.Лицензия = Новый ФиксированнаяСтруктура(Лицензия);

Результат.ОписаниеСоединения = Новый ФиксированнаяСтруктура(ОписаниеСоединения);

Результат.РабочийПроцесс = Новый ФиксированнаяСтруктура(РабочийПроцесс);

Данный метод вы можете применять на любой серверной базе, которая основана на библиотеке стандартных подсистем.

Теперь, переходим к части 2.

**Часть 2. Установка блокировок соединений и регламентных заданий.**

Теперь предлагаю рассмотреть еще одну полезную штуку - как установки блокировок соединений и регламентных заданий, так и регламентных заданий в частности.

Для понимания, установлена ли у нас блокировка регламентных заданий, воспользуемся функцией

**БлокировкаРегламентныхЗаданийИнформационнойБазы(Знач ПараметрыАдминистрированияКластера, Знач ПараметрыАдминистрированияИБ) Экспорт**,

которая вернет нам текущее состояние блокировки регламентных заданий для информационной базы.

Код выглядит вот так:

**Выясняем установлена ли блокировка регламентных заданий**

АдминКластера = АдминистрированиеКластера.ПараметрыАдминистрированияКластера();

АдминКластера.АдресАгентаСервера = "localhost";

АдминКластера.ИмяАдминистратораКластера = "admin";

АдминКластера.ПарольАдминистратораКластера= "77777";

АдминИБ = АдминистрированиеКластера

.ПараметрыАдминистрированияИнформационнойБазыКластера();

АдминИБ.ИмяВКластере = "ssl83test";

АдминИБ.ИмяАдминистратораИнформационнойБазы = "Администратор";

АдминИБ.ПарольАдминистратораИнформационнойБазы = "123456";

// состояние блокировки - у меня возвращает Истина

Блокировка = АдминистрированиеКластераCOM

.БлокировкаРегламентныхЗаданийИнформационнойБазы(АдминКластера, АдминИБ);

Заходим в консоль и смотрим, действительно "Да".

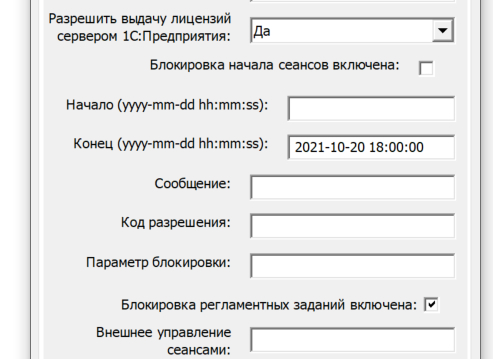


Рис.3.Блокировка регламентных заданий включена.

Теперь программно включим блокировку регламентных заданий, код будет вот такой:

**Снятие блокировки регламентных заданий**

АдминКластера = АдминистрированиеКластера.ПараметрыАдминистрированияКластера();

АдминКластера.АдресАгентаСервера = "localhost";

АдминКластера.ИмяАдминистратораКластера = "admin";

АдминКластера.ПарольАдминистратораКластера= "77777";

АдминИБ = АдминистрированиеКластера

.ПараметрыАдминистрированияИнформационнойБазыКластера();

АдминИБ.ИмяВКластере = "ssl83test";

АдминИБ.ИмяАдминистратораИнформационнойБазы = "Администратор";

АдминИБ.ПарольАдминистратораИнформационнойБазы = "123456";

// состояние блокировки - у меня возвращает Истина

Блокировка = АдминистрированиеКластераCOM

.БлокировкаРегламентныхЗаданийИнформационнойБазы(АдминКластера, АдминИБ);

Сообщить(Блокировка);

// снимаем блокировку регламентных заданий

АдминистрированиеКластераCOM

.УстановитьБлокировкуРегламентныхЗаданийИнформационнойБазы(

АдминКластера, АдминИБ, Ложь);

Блокировка = АдминистрированиеКластераCOM.БлокировкаРегламентныхЗаданийИнформационнойБазы(АдминКластера, АдминИБ);

// теперь, возвращает Ложь

Сообщить(Блокировка);

Возможность проверить состояние блокировки сеансов и регламентных заданий нам позволит функция

**БлокировкаСеансовИЗаданийИнформационнойБазы(**

**Знач ПараметрыАдминистрированияКластера,**

**Знач ПараметрыАдминистрированияИБ) Экспорт**.

Пример использования данной функции:

**Состояние блокировки сеансов и заданий информационной базы (Ложь или Истина)**

АдминКластера = АдминистрированиеКластера.ПараметрыАдминистрированияКластера();

АдминКластера.АдресАгентаСервера = "localhost";

АдминКластера.ИмяАдминистратораКластера = "quazare";

АдминКластера.ПарольАдминистратораКластера= "77777";

АдминИБ = АдминистрированиеКластера

.ПараметрыАдминистрированияИнформационнойБазыКластера();

АдминИБ.ИмяВКластере = "ssl83test";

АдминИБ.ИмяАдминистратораИнформационнойБазы = "Администратор";

АдминИБ.ПарольАдминистратораИнформационнойБазы = "123456";

//возвращает структуру по блокировкам

БлокировкаСеансовЗаданийСостояние = АдминистрированиеКластераCOM

.БлокировкаСеансовИЗаданийИнформационнойБазы(АдминКластера, АдминИБ);

Данный код вернет структуру по блокировкам такого вида:

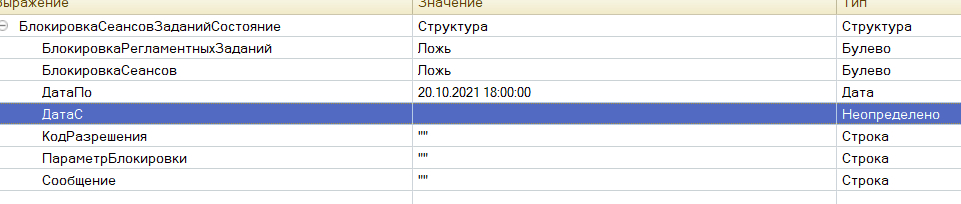


Рис.4.Возврат структуры по блокировкам базы данных.

Сейчас я "поиграюсь" с блокировками с помощью процедуры БСП

**УстановитьБлокировкуСеансовИЗаданийИнформационнойБазы(**

**Знач ПараметрыАдминистрированияКластера,**

**Знач ПараметрыАдминистрированияИБ,**

**Знач СвойстваБлокировкиСеансовИЗаданий) Экспорт**.

Напишу код, где запрещу все сеансы на один день, но дам код разрешения. Код выглядит вот так:

**Запрет сеансов до определенной даты с кодом разрешения**

АдминКластера = АдминистрированиеКластера.ПараметрыАдминистрированияКластера();

АдминКластера.АдресАгентаСервера = "localhost";

АдминКластера.ИмяАдминистратораКластера = "admin";

АдминКластера.ПарольАдминистратораКластера= "77777";

АдминИБ = АдминистрированиеКластера

.ПараметрыАдминистрированияИнформационнойБазыКластера();

АдминИБ.ИмяВКластере = "ssl83test";

АдминИБ.ИмяАдминистратораИнформационнойБазы = "Администратор";

АдминИБ.ПарольАдминистратораИнформационнойБазы = "123456";

БлокировкаСеансовЗаданийСостояние = АдминистрированиеКластераCOM

.БлокировкаСеансовИЗаданийИнформационнойБазы(АдминКластера, АдминИБ);

// выношу конструктор структуру а код процедуры

СтрБлок = Новый Структура("БлокировкаСеансов, ДатаС, ДатаПо,

Сообщение, КодРазрешения,ПараметрБлокировки,БлокировкаРегламентныхЗаданий");

СтрБлок.БлокировкаСеансов = истина;

СтрБлок.ДатаС = СтроковыеФункцииКлиентСервер.СтрокаВДату("20.10.2021");

СтрБлок.ДатаПо = СтроковыеФункцииКлиентСервер.СтрокаВДату("27.10.2021");

СтрБлок.Сообщение="Соединения с ИБ заблокировал";

СтрБлок.КодРазрешения="11";

СтрБлок.ПараметрБлокировки="";

СтрБлок.БлокировкаРегламентныхЗаданий=истина;

АдминистрированиеКластераCOM

.УстановитьБлокировкуСеансовИЗаданийИнформационнойБазы(

АдминКластера, АдминИБ, СтрБлок);

Заходим в консоль и видим, да - для этой базы все установлено (код писал 26.10.2021 - поэтому на один день).

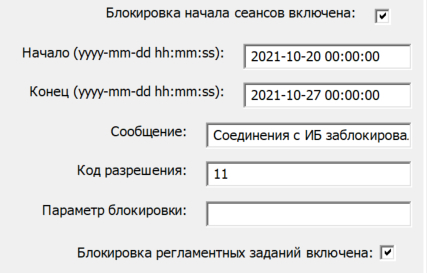


Рис.5.Программно заблокировал базу для соединений и регламентных заданий с кодом разрешения.

**Заключение и выводы**

В данной статье мы рассмотрели базовые и, я бы отметил, очень полезные приемы работы с кластером сервера с помощью библиотеки стандартных подсистем версии 3.1.5.208. Примеры методов вы можете использовать в любой стандартной конфигурации 1С, основой которой является БСП.

**Все эксперименты для написания этой статьи проводились на платформе 1С - 8.3.19.1264 и последней версии (на момент написания статьи) чистой конфигурации БСП 3.1.5.208. Использовалось СОМ-соединение.**