# Правила именования объектов

## Процедуры

### Процедуры вставки

Процедуры вставки данных начинаются с префикса «ins\_» далее имя таблицы (например ins\_Equipment);

### Процедуры редактирования

Процедуры редактирования данных начинаются с префикса «upd\_» далее имя таблицы (например upd\_Equipment);

### Процедуры удаления

Процедуры удаления данных начинаются с префикса «del\_» далее имя таблицы (например del\_Equipment);

### Процедура вычитки

Процедуры вычитки данных начинаются с префикса «get\_» далее имя таблицы (например get\_Equipment);

### Процедуры с output параметрами

Если процедура имеет output параметр, то имя параметра должно заканчиваться на Out.

## Представления

Представления начинаются с префикса «v\_» далее имя основной таблицы;

MS Entity Framework ставит требование что бы в таблицах и вью всегда был PK.

Поскольку в вью такое не возможно, то принимаем такое правило:

 - первое поле всегда это наш PK.

 - если по логике запроса первое поле уникально, то и все хорошо.

 - если по логике вью первое поле не уникально, то нужно создавать фиктивный PK, использовать newID() для заполнения этого фиктивного PK.

## Sequence

Sequence начинаются с префикса «gen\_» далее имя таблицы для которой он предназначен;

# Типы данных

## Дата+время

Для полей дата+время использовать тип datetimeoffset, а не datetime, поскольку RESTier не отливает datetime от date.

Если использование типа datetimeoffset невозможно (например KEP Server Data Logger не поддерживает тип datetimeoffset), то можно использовать тип datetime, но для вычитки данных с таблицы с этим полем, необходимо таблицу оформить в виде представления где поле с datetime преобразовать в datetimeoffset следующим образом:

ToDateTimeOffset([datetime\_field],0) as datetime\_field

## Hierarchyid

Нельзя использовать тип hierarchyidЮ поскольку он не поддерживается MS Entity Framework.

# Работа с Primary Key полями

Если ключевое поле таблицы имеет тип int, то его рекомендуется заполнять используя Sequence или Identity. При этом обязательно что бы Sequence был прописан как DEFAULT VALUE для поля :

Identity: ID INTEGER IDENTITY (1,1) PRIMARY KEY

Sequence: ALTER TABLE [dbo].[TableName] ADD DEFAULT (NEXT VALUE FOR [dbo].[gen\_SequenceName]) FOR [ID]

# Правила генерации пользовательских сообщение об ошибках

Для генерации пользовательского сообщения об ошибке необходимо использовать метод THROW.

Использовать RAISERROR запрещено.

Примеры.

Простое статическое сообщение:

THROW 60001, N'LENGTH param required', 1;

Сообщение с параметрами:

SET @err\_message = N'Заказ [' + CAST(@COMM\_ORDER AS NVARCHAR) + N'] уже существует';

THROW 60010, @err\_message, 1;

# Правила внесения изменений в структуры БД

Все изменения в структуре БД должны быть оформлены в виде SQL скриптов. Каждый такой скрипт должен содержать необходимый о достаточный набор кода для внесения изменений. Если изменение структуры требует изменение данных, то скрипт должен содержать и код по модификации этих данных.

Все скрипты должны сохраняться в папке service\_packs в репозитории <https://github.com/arcelormittalkriviyrih/isa95_database>

Порядок выполнения скриптов – согласно сортировки по имени файла.

Примечание: Файлы скриптов должны быть в кодировке UTF-8/

## Правила именования скриптов

<year>\_<month>\_<day>\_<n>\_<short\_name>.sql

Где:

year – год создания скрипта

month – месяц создания скрипта

day – день создания скрипта

n – номер скрипта за день

short\_name – короткое имя скрипта, кратко описывающее суть изменения

Имя скрипта не должно содержать пробелов.

*Например*: 2016\_04\_10\_1\_add\_equipement\_table.sql

## Правила написания скриптов

* Каждый скрипт должен заканчиваться вызовом GO
* Если скрипт вставляет данные, но перед вставкой необходимо выполнять проверку наличия этих данных. Скрипт может быть вызван несколько раз (например, в скрипте была ошибка), поэтому во избежание проблемы с дубликатами данных, необходимо проверять что таких данных нет в таблице
* Если скрипт изменяет структуры, то перед этим необходимо проверять что эти изменения уже не были выпилены, например, проверять что добавляемое поле в таблицу уже еще не существует.

## Применение скриптов

Скрипты, сохраненные в репозитории, автоматически выполняются систему Jenkins при вызове команды на сборку новой версии.

Успешно примененные скрипты регистрируются в таблице ServicePacksFiles.

Если скрипт не выполнился успешно, то ошибки его выполнения необходимо смотреть в логах сборки соответствующего билда в Jenkins

После исправления скрипта, его необходимо обновить в репозитории и вызвать сборку новой версии повторно.

**Внимание:** Если скрипт уже был успешно выполнен, то его запрещено изменять, поскольку эти изменения не будет применены при следующей сборке.

# Настройка IIS

Для решения проблемы с Impersonation необходимо:

1. настроить aspnet.config файл (находится в папке C:\Windows\Microsoft.Net\Framework64\v4.0.30319)

<legacyImpersonationPolicy enabled="false"/>

<alwaysFlowImpersonationPolicy enabled="true"/>

Больее детально - <https://blogs.msdn.microsoft.com/tom/2008/04/22/making-an-asynchronous-call-using-the-impersonation-identity/>

1. В настройках веб сервиса в разделе Authentication:
   1. Оставить включенными только ASP.NET Impersonation, Windows Authentication.
   2. Для Windows Authentication в Advanced Settings снять галочку с Enable Kernel-mode authentication
   3. Для Windows Authentication в Providers оставить только протокол Negotiate:Kerberos
2. Application pool для веб сервиса должен быть запущен от имени учетной записи которая:
   1. Имеет права доступа к базе на получение списка всех таблиц, вбюх и процедур
   2. Имеет достаточные права для делегирования учетных записей для работы Kerberos (<https://blogs.msdn.microsoft.com/chiranth/2014/04/17/setting-up-kerberos-authentication-for-a-website-in-iis/>)

# Jenkins

## Установка и настройка Jenkins

1. Скачать Jenkins c сайта <https://jenkins.io/>

Weekly Release , версия для Windows

1. Установить Jenkins
2. После установки Jenkins доступен по адресу <http://localhost:8080/>
3. На данный момент доступен <http://krr-tst-palbp01:8080/>
4. Установить следующие Plugins:

MSTestRunner plugin

Active Directory plugin

Build Name Setter Plugin

Credentials Binding Plugin

Doxygen Plug-in

Git client plugin

Git plugin

Git server plugin

GitHub API plugin

GitHub Branch Source Plugin

GitHub Organisation Folder Plugin

GitHub plugin

MSBuild Plugin

Warnings Plug-in

Workspace Cleanup Plugin

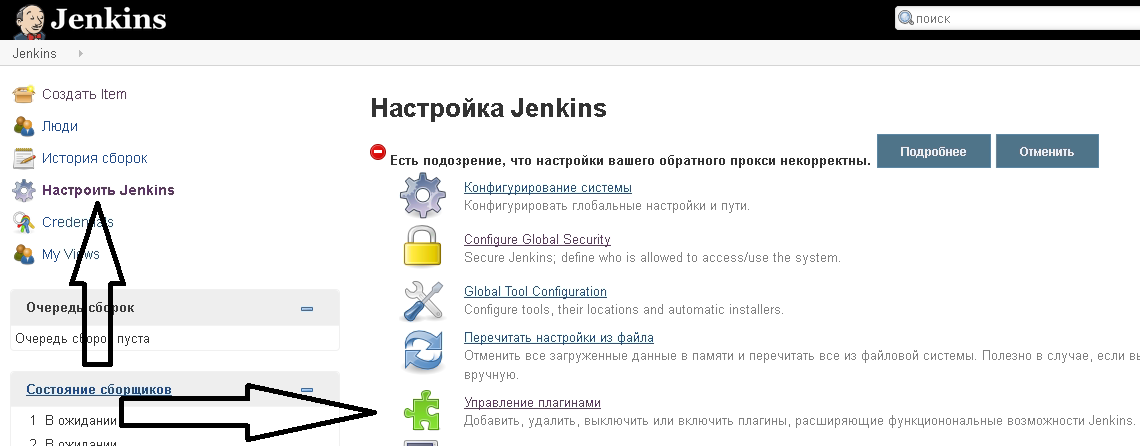
MSTest plugin

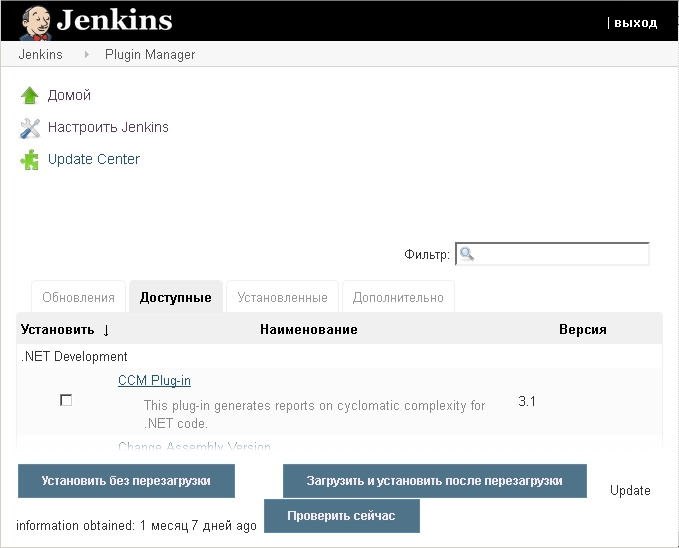
Plugins можно установить при инициализации (первом запуске) Jenkins или

На странице Управление плагинами в Jenkins:

Jenkins 🡪 Настроить Jenkins(Manage Jenkins) 🡪 Управление плагинами (Manage Plugins)

Вкладка Доступные(Available)

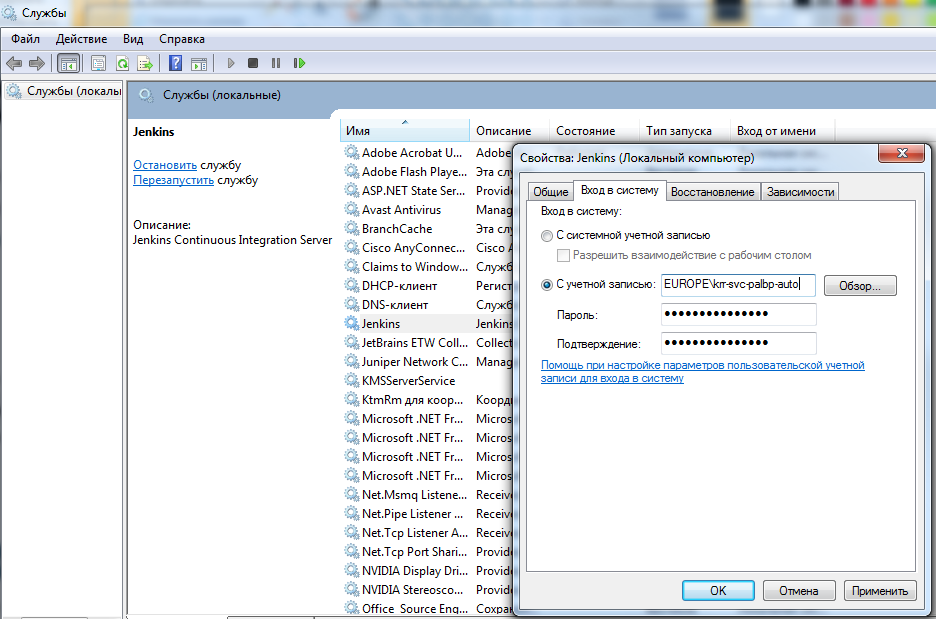




1. Установить учётную запись EUROPE\krr-svc-palbp-auto для службы Jenkins:

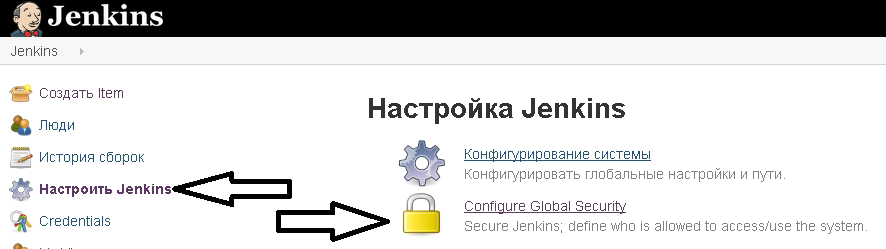
Панель управления🡪Администрирование🡪Службы

Выбрать Jenkins(двойной клик) , Вкладка вход в систему , указать учётную запись EUROPE\krr-svc-palbp-auto

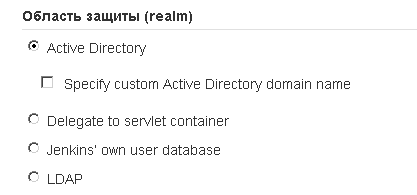


1. Настроить политику безопасности Jenkins:

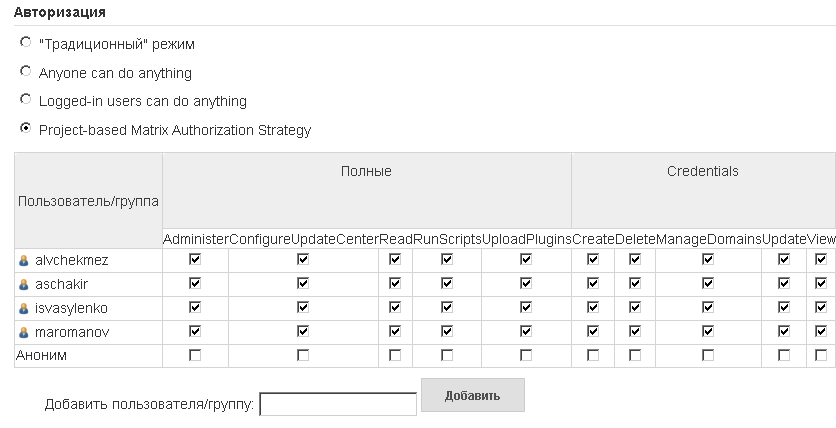
Jenkins 🡪 Настроить Jenkins(Manage Jenkins) 🡪Configure Global Security



Установить область защиты Active Directory



Установить Авторизацию Project-based Matrix Authorization Strategy и раздать права пользователям



### Создание папки ArcelorMittal для задач

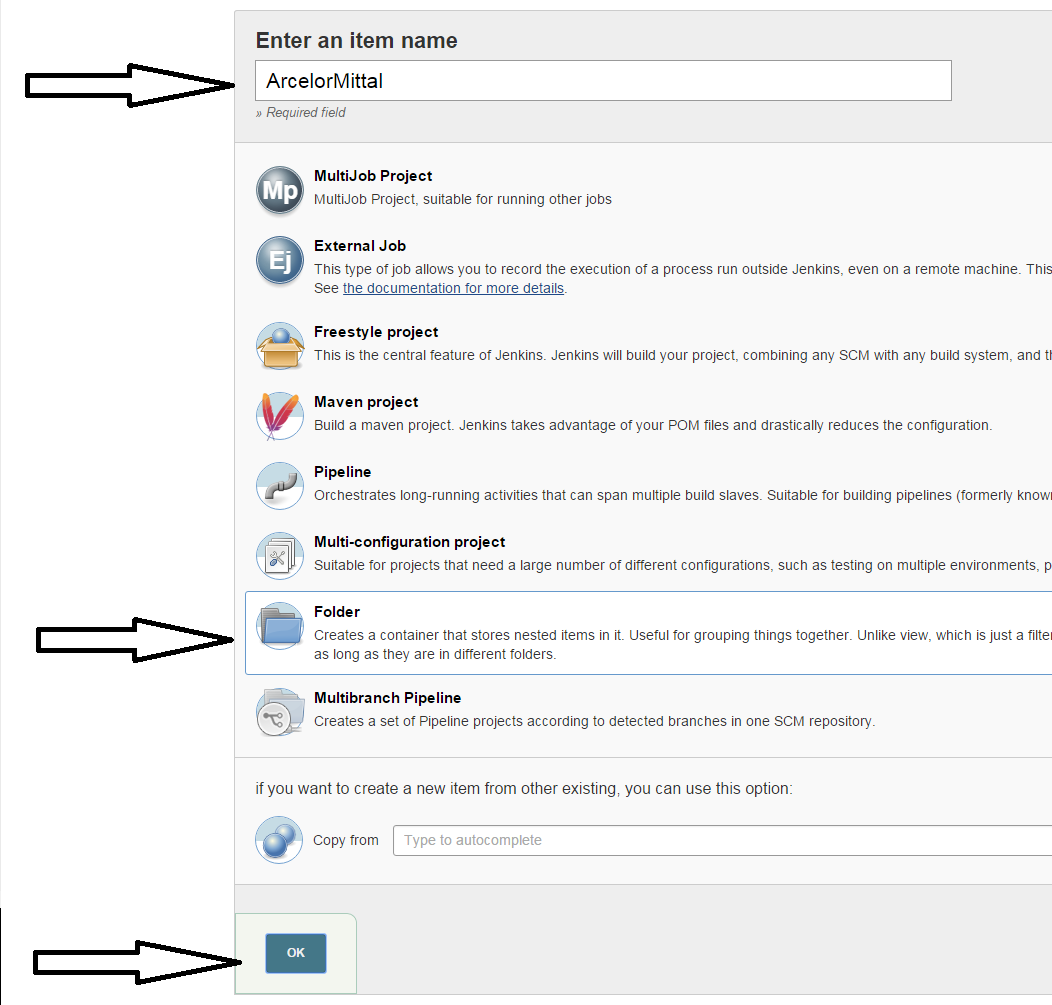
Пункт меню создать Item

Ввести имя

Выбрать тип Folder

Нажать ОК

Нажать Save



### Создание задачи isa95\_database

В папке ArcelorMittal создать задачу обновления базы данных isa95\_database:

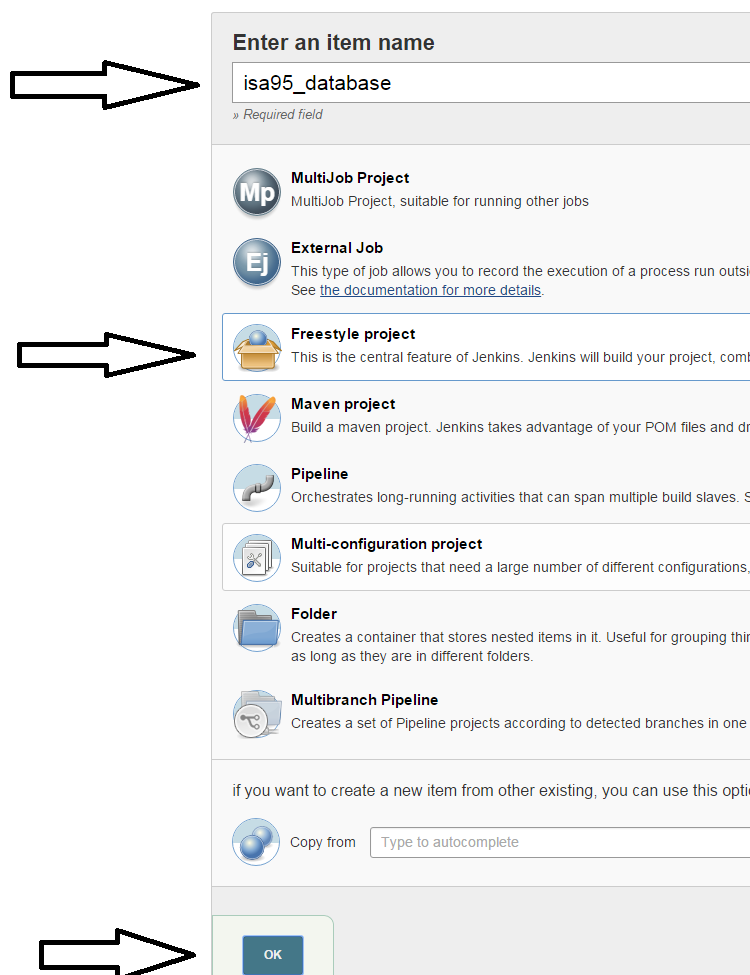
Перейти в папку ArcelorMittal

Выбрать Создать Item

Ввести имя

Выбрать тип Item - Создать задачу со свободной конфигурацией (Freestyle project)

Нажать Ок



Установить чекбоксы:

- GitHub project

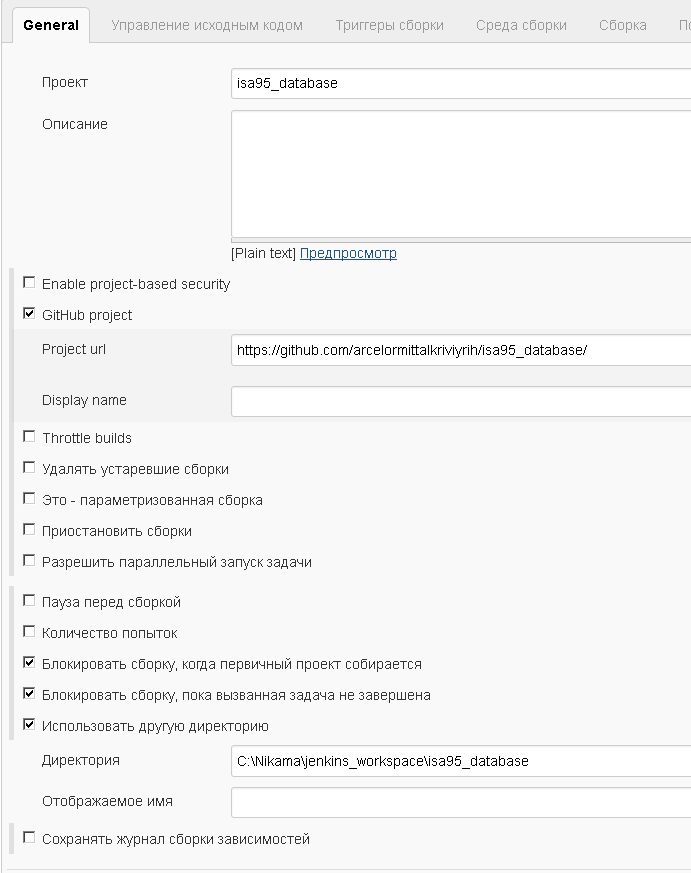
Project url: <https://github.com/arcelormittalkriviyrih/isa95_database/>

- Блокировать сборку, когда первичный проект собирается

- Блокировать сборку, пока вызванная задача не завершена

- Использовать другую директорию

Директория: C:\Nikama\jenkins\_workspace\isa95\_database



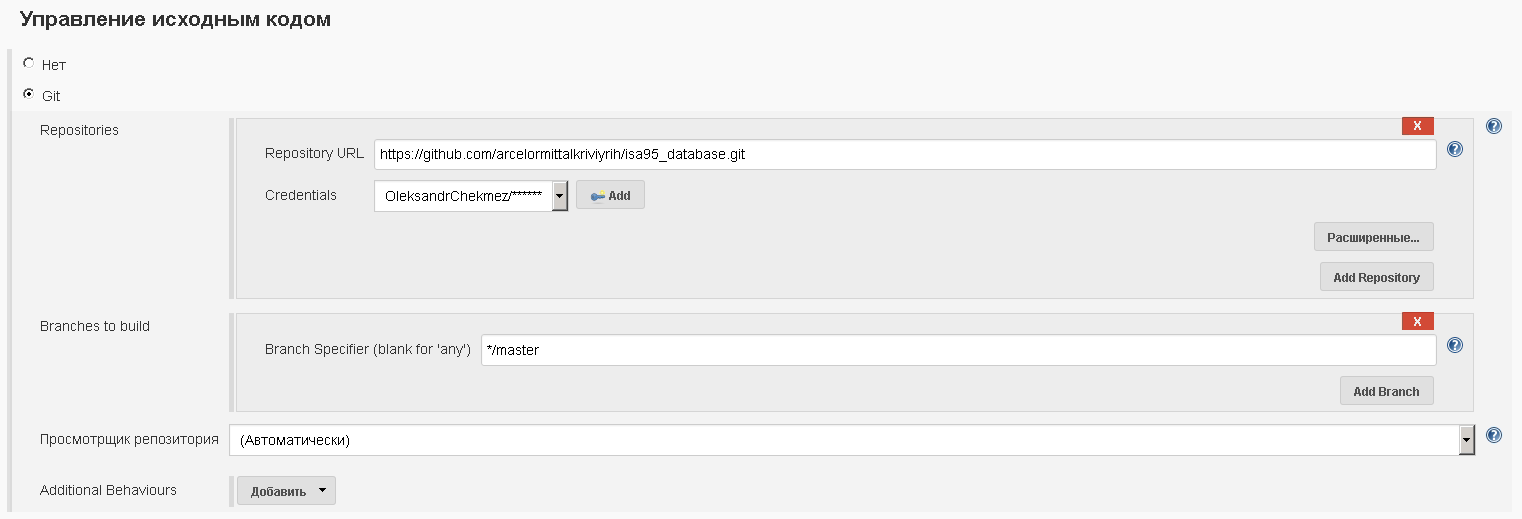
Управление исходным кодом:

Установить Git

Repository URL: <https://github.com/arcelormittalkriviyrih/isa95_database.git>

Добавить учётную запись для доступа к GitHub

Branch Specifier: \*/master



Триггеры сборки: Оставить неотмеченным

Среда сборки:

Отметить Delete workspace before build starts

Сборка:

Добавить шаг сборки

Выполнить команду Windows



Указать правильный путь к файлу create\_test\_db\_from\_prod\_dump.bat

Добавить шаг сборки

Выполнить команду Windows

Указать правильный путь к файлу apply\_service\_packs.bat

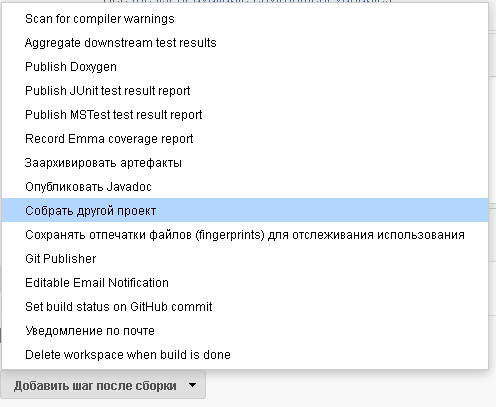
Файлы create\_test\_db\_from\_prod\_dump.bat и apply\_service\_packs.bat доступны в GitHub

<https://github.com/arcelormittalkriviyrih/isa95_database/tree/master/jenkins_scripts>



Послесборочные операции:

Добавить шаг после сборки🡪Собрать другой проект



Указать имя odata\_unified\_svc

Нажать сохранить

Вернуться к папке ArcelorMittal

### Создание задачи odata\_unified\_svc

В папке ArcelorMittal создать задачу обновления базы данных odata\_unified\_svc:

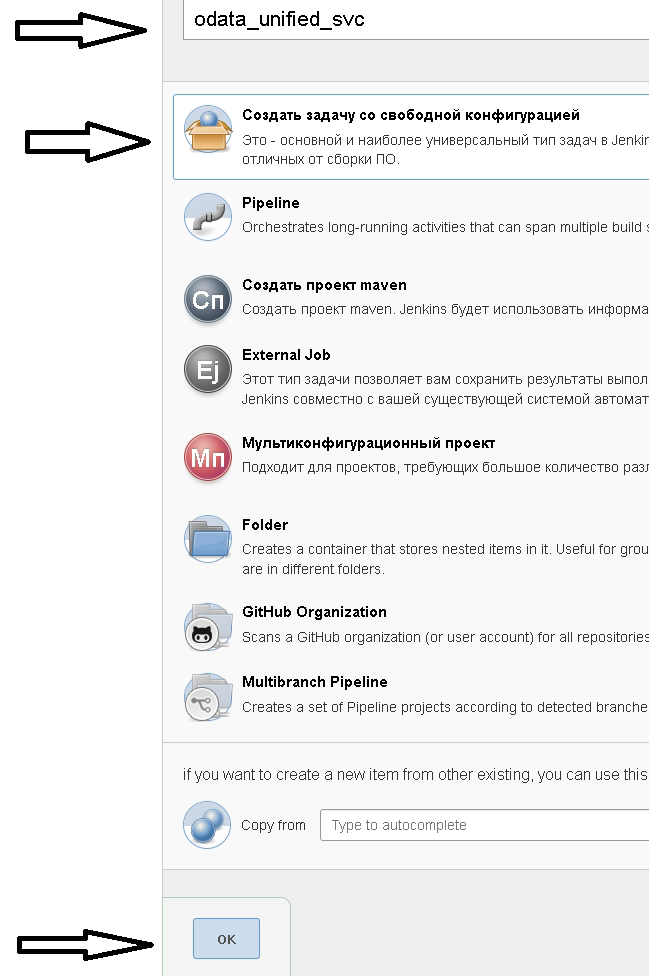
Перейти в папку ArcelorMittal

Выбрать Создать Item

Ввести имя

Выбрать тип Item - Создать задачу со свободной конфигурацией (Freestyle project)

Нажать Ок



Установить чекбоксы:

- GitHub project

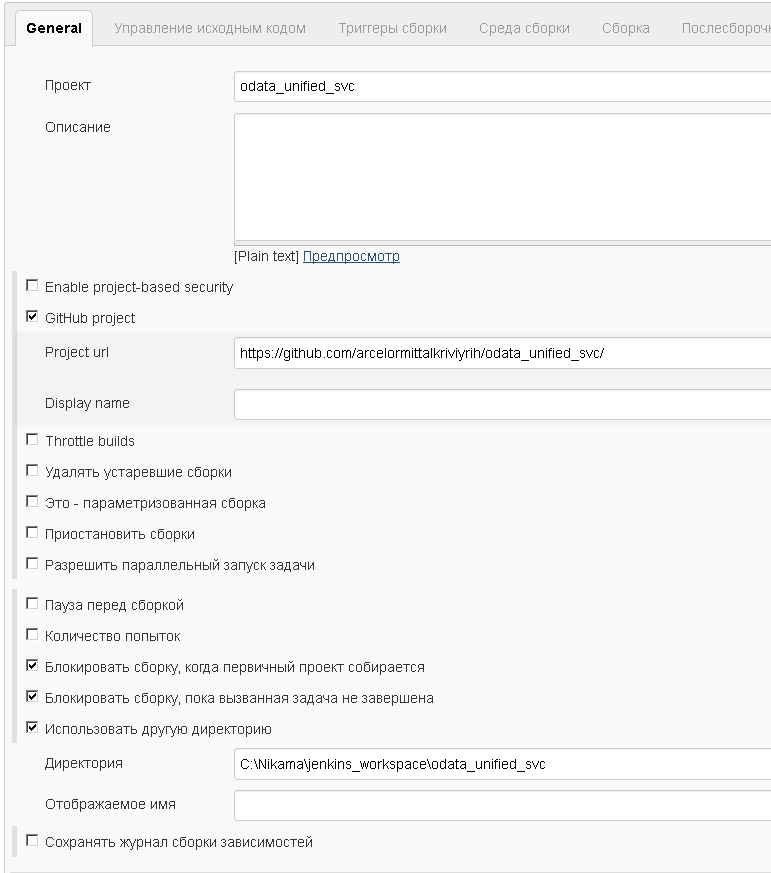
Project url: https://github.com/arcelormittalkriviyrih/odata\_unified\_svc/

- Блокировать сборку, когда первичный проект собирается

- Блокировать сборку, пока вызванная задача не завершена

- Использовать другую директорию

Директория: C:\Nikama\jenkins\_workspace\odata\_unified\_svc



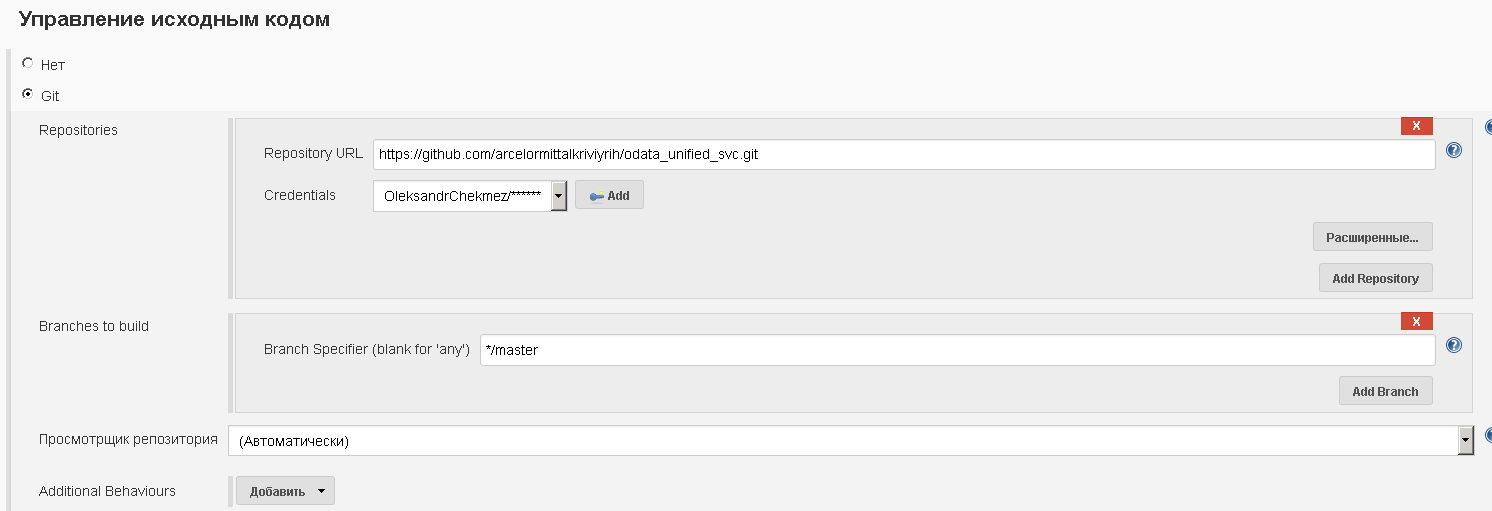
Управление исходным кодом:

Установить Git

Repository URL: https://github.com/arcelormittalkriviyrih/odata\_unified\_svc.git

Добавить учётную запись для доступа к GitHub

Branch Specifier: \*/master



Триггеры сборки: Оставить неотмеченным

Среда сборки:

Отметить Delete workspace before build starts

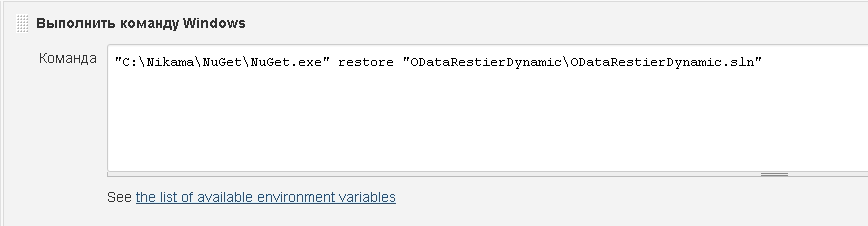
Сборка:

Добавить шаг сборки

Выполнить команду Windows



Добавить комманду: "C:\Nikama\NuGet\NuGet.exe" restore "ODataRestierDynamic\ODataRestierDynamic.sln"



**Важно!:** Необходима установленная утилита NuGet. Утилиту можно скачать по ссылке <https://dist.nuget.org/index.html>

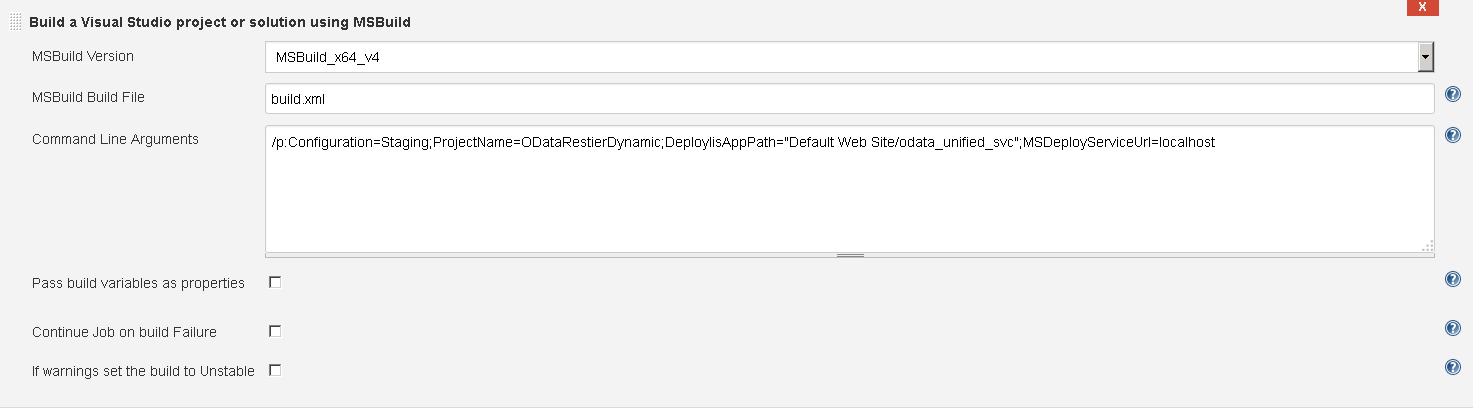
Добавить шаг сборки: Build a Visual Studio project or solution using MSBuild

Установить

MSBuild Version - MSBuild\_x64\_v4

MSBuild Build File - build.xml

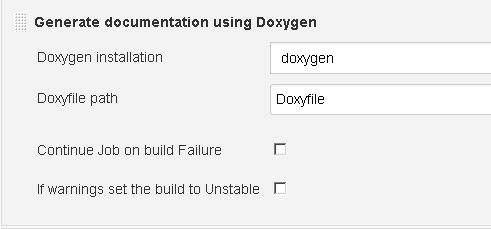
Command Line Arguments - /p:Configuration=Staging;ProjectName=ODataRestierDynamic;DeployIisAppPath="Default Web Site/odata\_unified\_svc";MSDeployServiceUrl=localhost



Добавить шаг сборки: Generate documentation with Doxygen

Doxygen installation - doxygen

Doxyfile path – Doxyfile

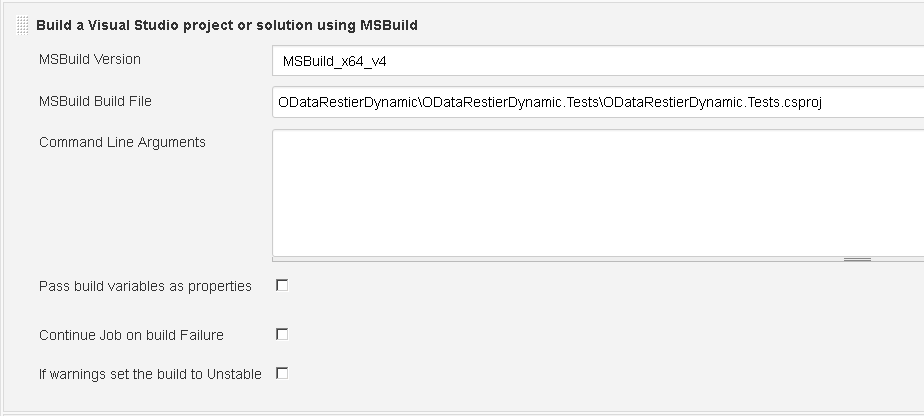


Добавить шаг сборки: Build a Visual Studio project or solution using MSBuild

Установить

MSBuild Version - MSBuild\_x64\_v4

MSBuild Build File - ODataRestierDynamic\ODataRestierDynamic.Tests\ODataRestierDynamic.Tests.csproj



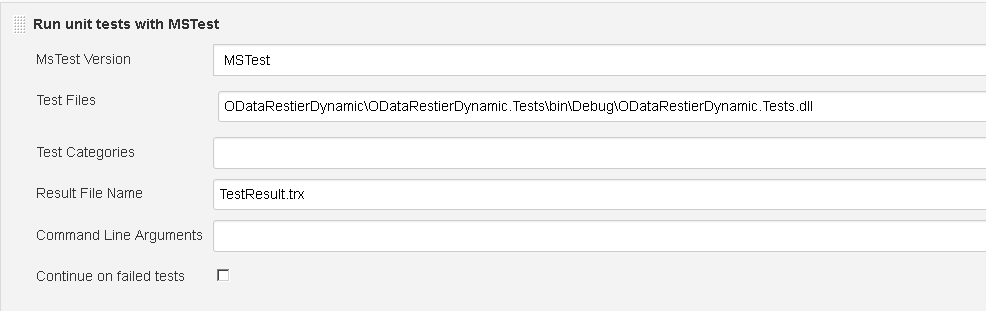
Добавить шаг сборки: Run unit tests with MSTest

Установить

MsTest Version – MSTest

Test Files - ODataRestierDynamic\ODataRestierDynamic.Tests\bin\Debug\ODataRestierDynamic.Tests.dll

Result File Name - TestResult.trx

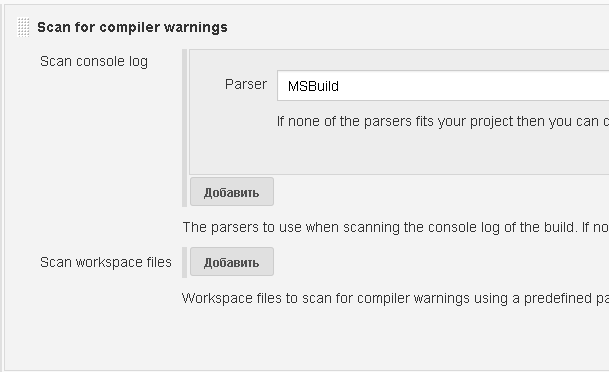


Послесборочные операции:

Добавить шаг после сборки: Scan for compiler warnings

Установить

Scan console log 🡪 Parser – MSBuild

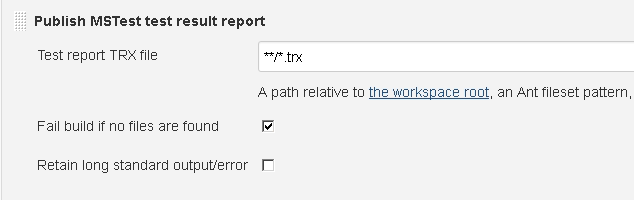


Добавить шаг после сборки: Publish MSTest test result report

Установить

Test report TRX file - \*\*/\*.trx

Fail build if no files are found – вкл



Добавить шаг после сборки🡪Собрать другой проект

Указать имя odata\_unified\_frontend



Нажать сохранить

Вернуться к папке ArcelorMittal

### Создание задачи odata\_unified\_frontend

В папке ArcelorMittal создать задачу обновления базы данных odata\_unified\_frontend:

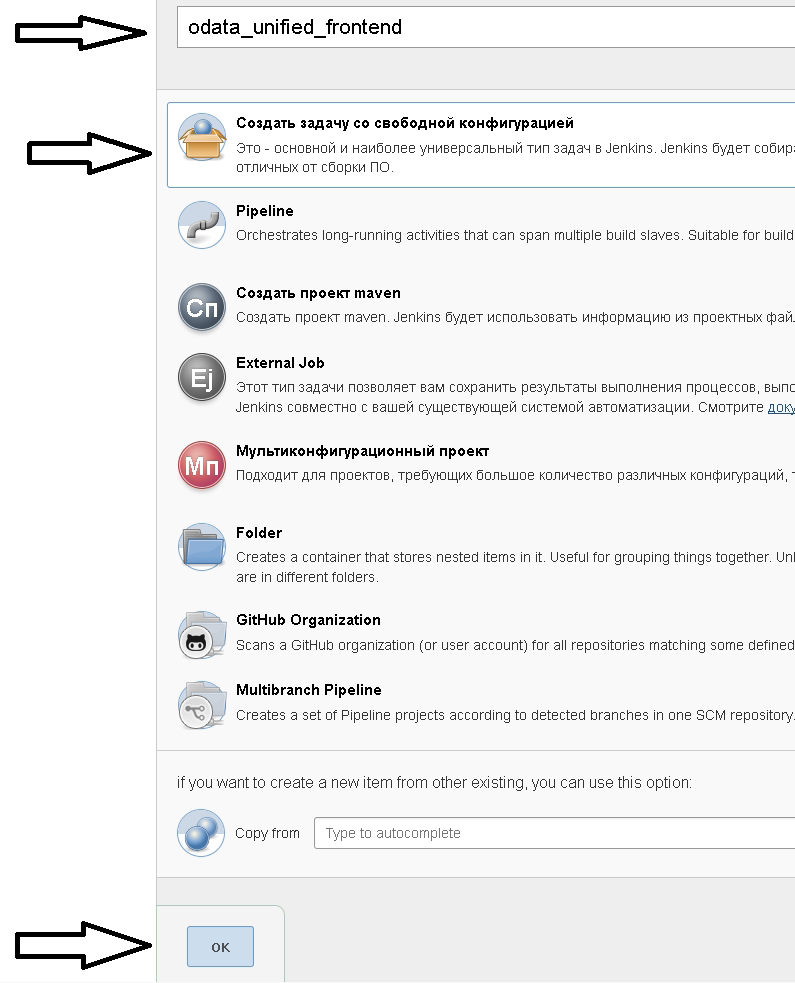
Перейти в папку ArcelorMittal

Выбрать Создать Item

Ввести имя

Выбрать тип Item - Создать задачу со свободной конфигурацией (Freestyle project)

Нажать Ок



Установить чекбоксы:

- GitHub project

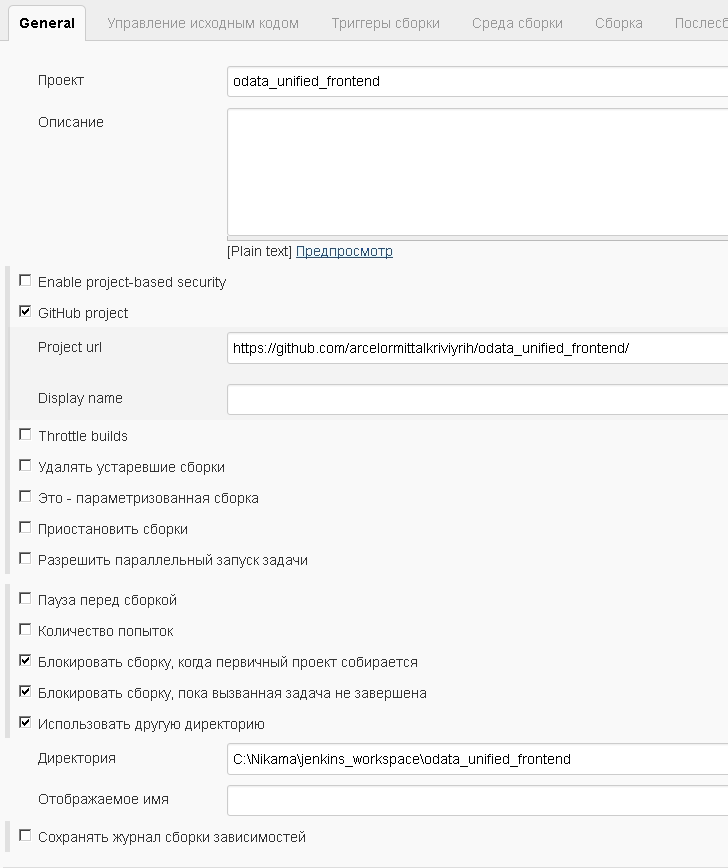
Project url: <https://github.com/arcelormittalkriviyrih/odata_unified_frontend/>

- Блокировать сборку, когда первичный проект собирается

- Блокировать сборку, пока вызванная задача не завершена

- Использовать другую директорию

Директория: C:\Nikama\jenkins\_workspace\odata\_unified\_frontend



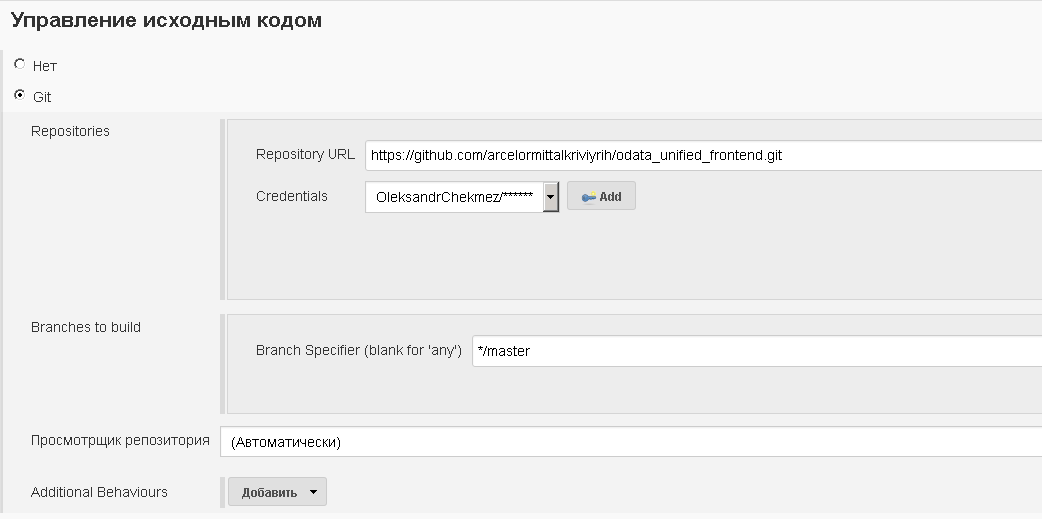
Управление исходным кодом:

Установить Git

Repository URL: https://github.com/arcelormittalkriviyrih/odata\_unified\_frontend.git

Добавить учётную запись для доступа к GitHub

Branch Specifier: \*/master



Триггеры сборки: Оставить неотмеченным

Среда сборки:

Отметить Delete workspace before build starts

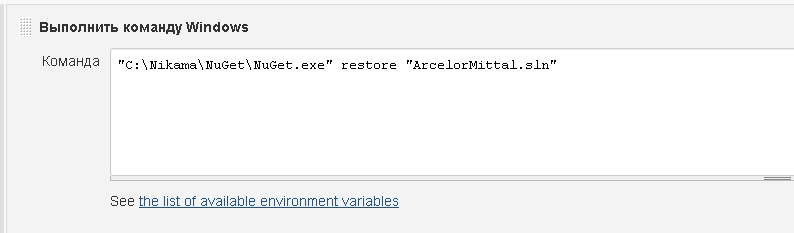
Сборка:

Добавить шаг сборки

Выполнить команду Windows



Добавить комманду: "C:\Nikama\NuGet\NuGet.exe" restore "ArcelorMittal.sln"



**Важно!:** Необходима установленная утилита NuGet. Утилиту можно скачать по ссылке <https://dist.nuget.org/index.html>

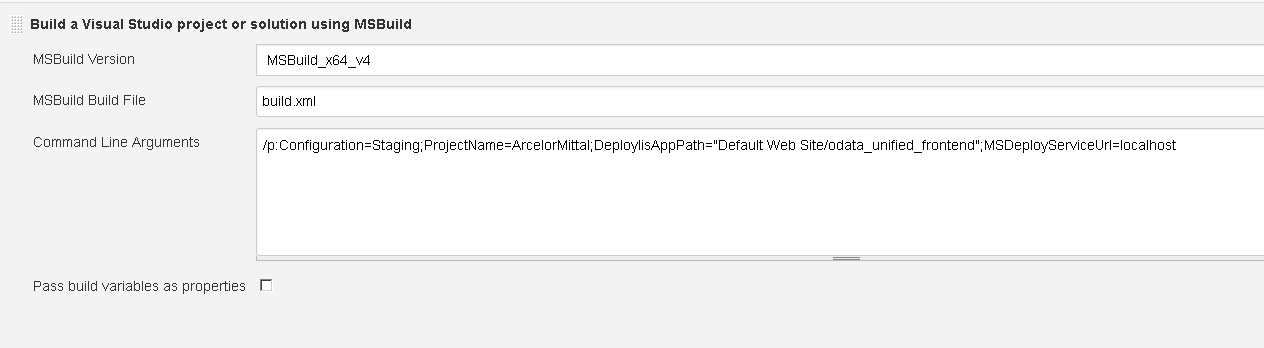
Добавить шаг сборки: Build a Visual Studio project or solution using MSBuild

Установить

MSBuild Version - MSBuild\_x64\_v4

MSBuild Build File - build.xml

Command Line Arguments - /p:Configuration=Staging;ProjectName=ArcelorMittal;DeployIisAppPath="Default Web Site/odata\_unified\_frontend";MSDeployServiceUrl=localhost

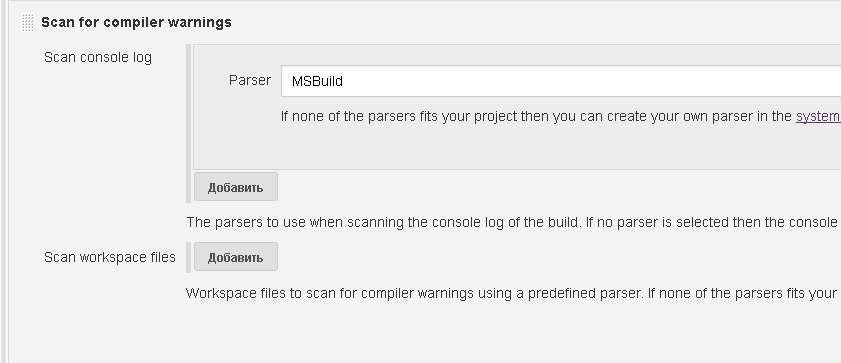


Послесборочные операции:

Добавить шаг после сборки: Scan for compiler warnings

Установить

Scan console log 🡪 Parser – MSBuild

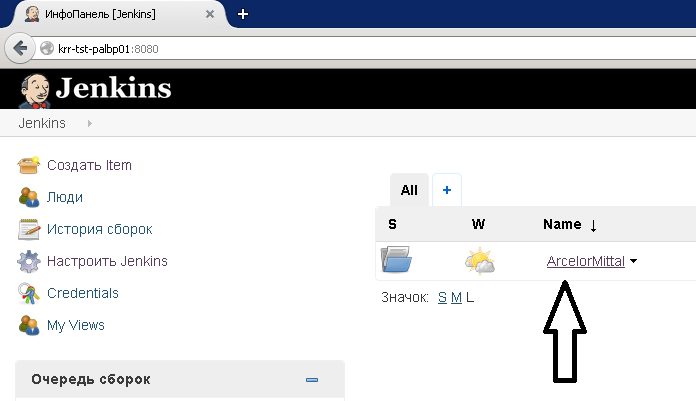


Нажать сохранить

## Использование Jenkins

### Запуск обновления проекта

1. Открыть Jenkins (Сейчас доступен по ссылке <http://krr-tst-palbp01:8080/)>
2. Открыть папку [ArcelorMittal](http://krr-tst-palbp01:8080/job/ArcelorMittal/)



1. Запланировать сборку isa95\_database



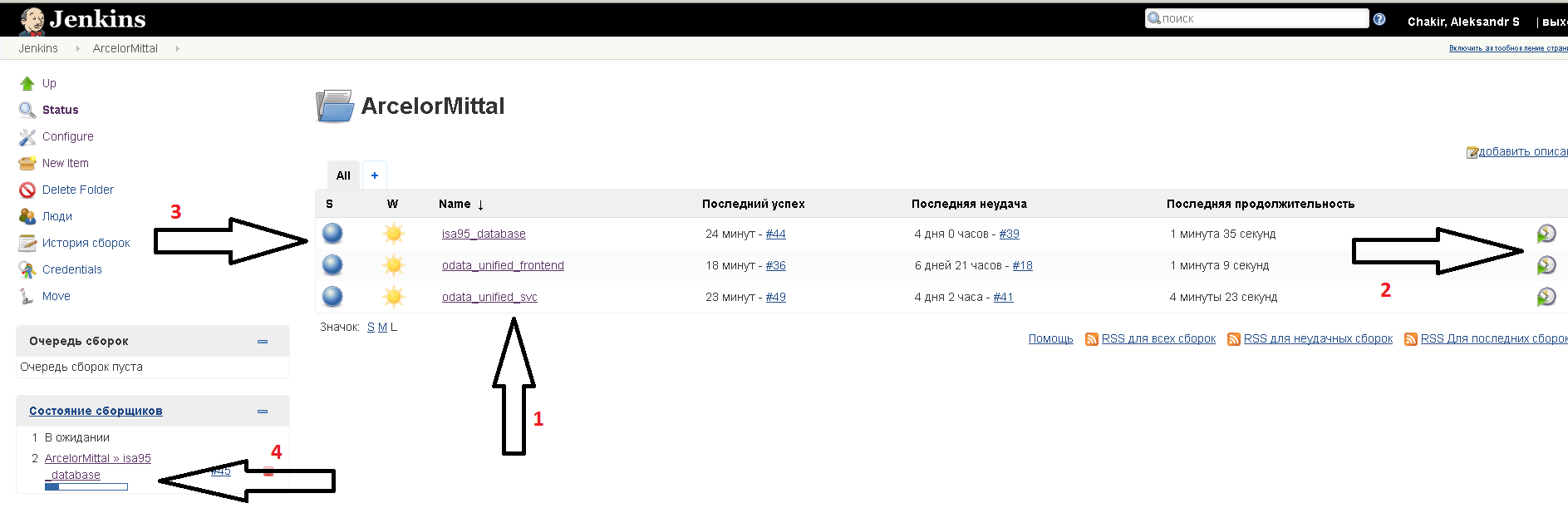
При успешной сборке isa95\_database задача odata\_unified\_svc запустится автоматически.

При успешной сборке odata\_unified\_svc задача odata\_unified\_frontend запустится автоматически.

### Страница папки [ArcelorMittal](http://krr-tst-palbp01:8080/job/ArcelorMittal/)

На странице папки ArcelorMittal доступно:

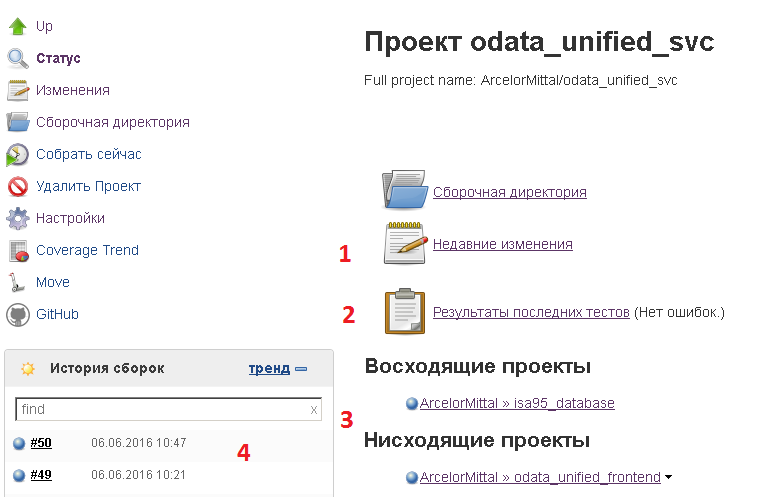
1. Список задач для сборки. Для открытия задачи необходимо кликнуть по имени
2. Запуск сборки
3. Статусы последних сборок (Синий – успешно , Красный -неуспешно)
4. Состояние сборщиков



### Страница Задачи

На странице задачи доступно

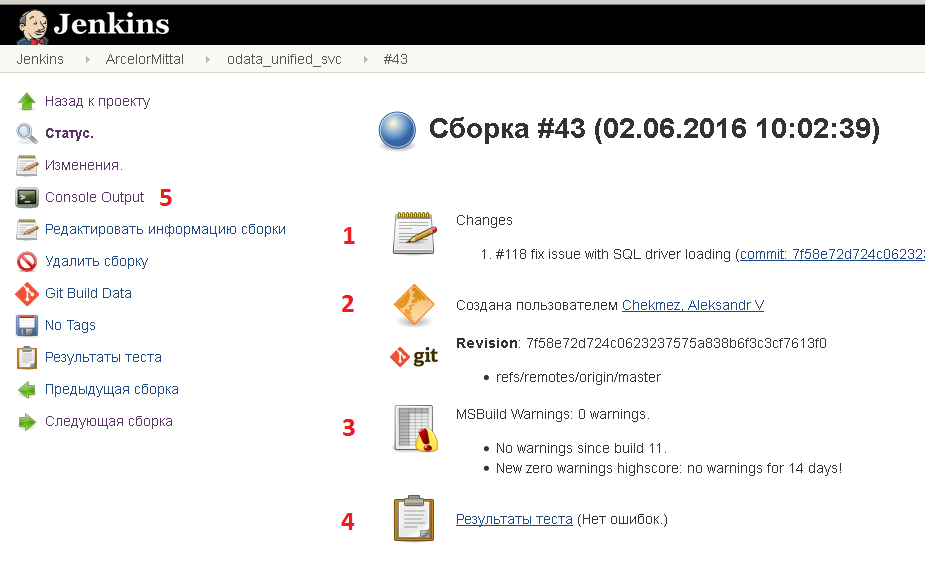
1. Последние изменения
2. Результаты последних тестов
3. Связанные задачи
4. История сборок. Для открытия сборки необходимо кликнуть по номеру



### Страница сборки

На странице сборки доступно:

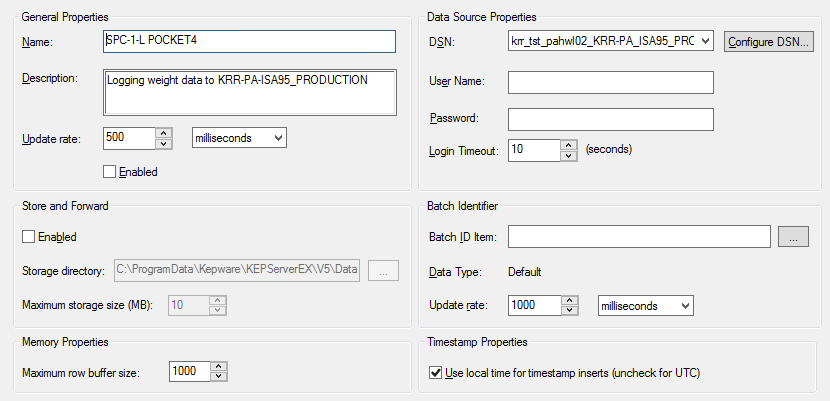
1. Изменения вошедшие в сборку
2. Кем запущена
3. Ошибки и предупреждения MSBuild
4. Результаты тестов
5. Log сборки



# Настройка Data Logger в KEP Server

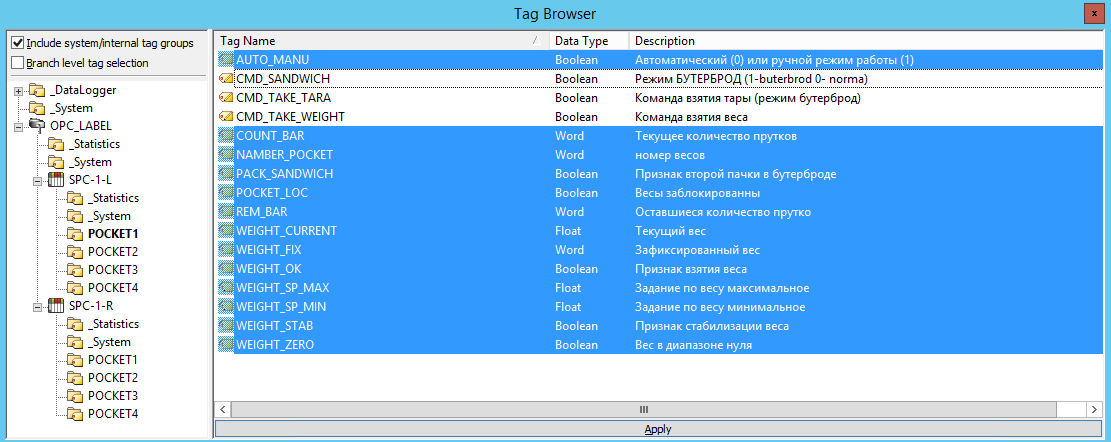
Для каждого весов/кармана необходимо создавать новую Log Group.

На закладке General необходимо указать имя и DSN подключение к базе:



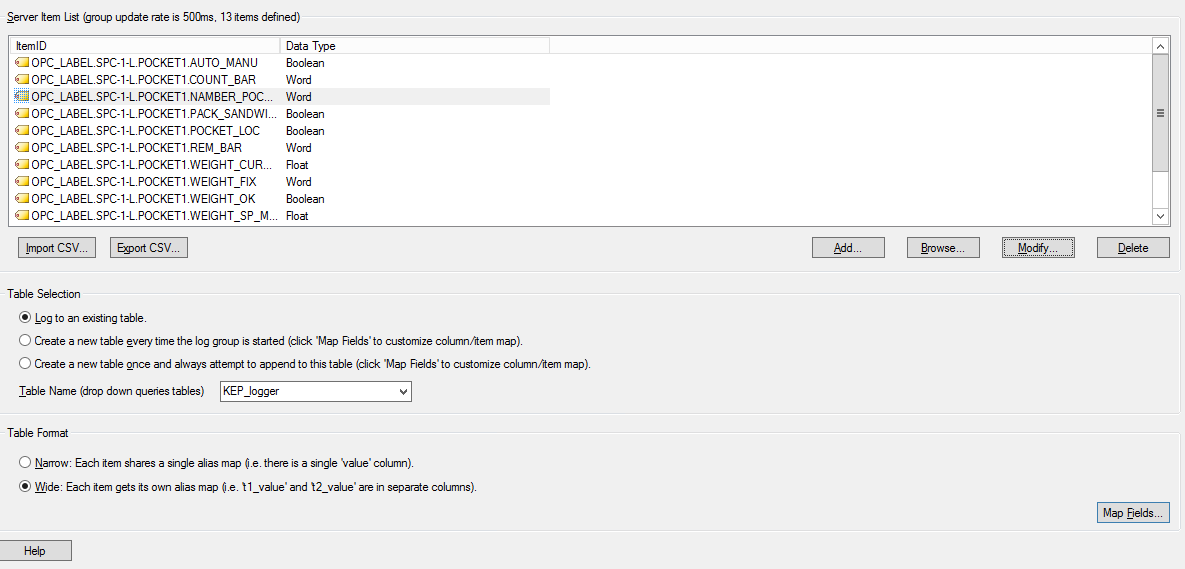
На закладке Data Map указываем что и как записывать в базу.

В рамках одной Log Group необходимо добавить все теги, с соответствующего устройства, что нужно записывать в базу:



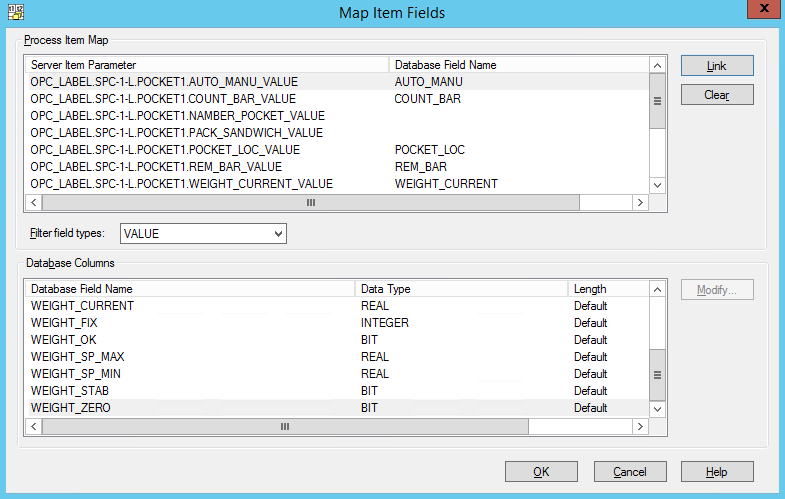
Примечание: теги что не пишем в базу, добавлять не нужно (теги команд)

Указать таблицу в какую записывать данные и формат Wide:



Нажать на кнопку Map Fields и указать какие теги в какие поля таблицы записывать:

|  |  |
| --- | --- |
| **Имя тега** | **Поле в таблице KEP\_Logger** |
| AUTO\_MANU\_VALUE | AUTO\_MANU |
| COUNT\_BAR\_VALUE | COUNT\_BAR |
| NAMBER\_POCKET\_VALUE | NAMBER\_POCKET |
| PACK\_SANDWICH\_VALUE | PACK\_SANDWICH |
| POCKET\_LOC\_VALUE | POCKET\_LOC |
| REM\_BAR\_VALUE | REM\_BAR |
| WEIGHT\_CURRENT\_TIMESTAMP | TIMESTAMP |
| WEIGHT\_CURRENT\_VALUE | WEIGHT\_CURRENT |
| WEIGHT\_FIX\_VALUE | WEIGHT\_FIX |
| WEIGHT\_OK\_VALUE | WEIGHT\_OK |
| WEIGHT\_SP\_MAX\_VALUE | WEIGHT\_SP\_MAX |
| WEIGHT\_SP\_MIN\_VALUE | WEIGHT\_SP\_MIN |
| WEIGHT\_STAB\_VALUE | WEIGHT\_STAB |
| WEIGHT\_ZERO\_VALUE | WEIGHT\_ZERO |



На закладке Triggers указываем правило когда записывать данные:

# Выполнение действий от имени другого пользователя

Если пользователь не имеет права выполнить определенное действие (например ручной ввод веса), то это действие может быть разрешено другому пользователю. Что бы подключится к системе от имени другого пользователя, при этом не выходить с Windows, можно запустить Internet Explorer от имени другого пользователя. Для этого удобней всего воспользоваться официальной утилитой Microsoft - ShellRunas (<https://technet.microsoft.com/en-us/sysinternals/shellrunas.aspx> )

Для простоты и удобства можно создать ярлык, который будет выполнять следующую команду:

«<path to utility>\shellrunas.exe "C:\Program Files\Internet Explorer\iexplore.exe" [http://krr-tst-palbp01/odata\_unified\_frontend/#/ru/marker»](http://krr-tst-palbp01/odata_unified_frontend/#/ru/marker)

Таким образом будет запрошен логин и пароль от имени которого запуститься Internet Explorer и откроется указанная страница.