



# Plansoft.org: Integracja z BAZUS

@Autor Maciej Szymczak

@wersja 2022.12

Opis funkcjonalny .....	1
Wprowadzenie danych w BAZUS .....	2
Relacje pomiędzy grupami .....	3
Pobranie danych do Plansoft.org- harmonogram .....	3
Jak sprawdzić, czy przesyłanie danych działa? .....	3
Planujemy zajęcia! .....	4
Przesyłamy rozkład do BAZUS .....	4
Raport: Co nie zostało wysłane do Bazusa? .....	5
Dla informatyków: szczegóły techniczne .....	5
Jak to zostało zrobione? .....	5
Przesłanie danych słownikowych .....	6
Przesłanie planu studiów .....	6
Odesłanie ułożonego rozkładu zajęć .....	7
Bazus_sub_map .....	7
Zastosowany Oprogramowanie .....	8
Instalacja połączenia .....	8
Rozwiązywanie problemów .....	11
Po co jest parametr pCleanpMode w pakiecie Integration? .....	12
ORA-00001: unique constraint (PLANNER.ROOM_UK) violated .....	12
ORA-00001: unique constraint (PLANNER.SUB_ABBREVIATION_I) violated .....	14
ORA-00001: unique constraint (PLANNER.GRO_ABBREVIATION_I) violated .....	14
Przedmioty zawierające znaki końca wiersza w nazwach .....	14
ORA-00001: unique constraint (PLANNER.SUB_NAME_UI) violated .....	15
ORA-00001: unique constraint (PLANNER.LEC_NAME_UI) violated .....	15
Scalanie istniejącego przedmiotu z nowym przedmiotem .....	15
Sprawdzenie, które pozycje z planu zajęć nie mogą być zaimportowane .....	16

## Opis funkcjonalny

W celu uruchomienia integracji z systemem Bazus konieczne jest uruchomienie funkcji integracyjnych w systemie Bazus, szczegóły należy uzgodnić z firmą Simple, dostawcą systemu Bazus.



Integracja opisana w tym dokumencie:

1. Przesyła dane słownikowych oraz plan studiów z BAZUS do Plansoft.org.
2. Następnie, po zaplanowaniu zajęć, przesyła gotowy rozkład zajęć do systemu BAZUS.



Ten rozdział opisuje czynności, jakie wykonujemy w kolejności chronologicznej.

### Wprowadzenie danych w BAZUS

Następujące dane muszą zostać wprowadzone w systemie BAZUS:

- Słowniki: wykładowcy, grupy, sale, przedmioty, formy prowadzenia zajęć.
  - **Integration id.** Każdy rekord posiada tak zwany integration\_id, jest to unikatowy, stały identyfikator rekordu, który jest nadawany przez system BAZUS. Integration id jest generowany automatycznie, nie musimy się o niego troszczyć. Po przesłaniu danych ten numer możemy zobaczyć w Plansoft.org tutaj.

Wykładowcy

Sortuj wg

Skrót

Dowolna fraza

>> 81897

zajęcia

Finanse

Dostęp

Jedn. org.

Skrót	Kolor	Tytuł	Imię	Nazwisko	Nazwa jedn. org.	Kod struktury jedn. org.	Przedmioty	Słowa kluczowe	Integration id
AdAr	dr inż.	ADAMCZYK	Arkadiusz	Wojkowska Akademia WAT		#przedmiot	#sk	8189774	
ApJa	mgr inż.	APANOWICZ	Jarosław	Wojkowska Akademia WAT				8189776	
BeJa	mgr inż.	BEDNARSKI	Jakub	Wojkowska Akademia WAT				8189778	

Możemy wyszukiwać rekordy za pomocą integration\_id wpisując nr w polu Dowolna fraza.

Jeżeli numer jest pusty to znaczy, że rekord nie był importowany z BAZUS, lecz został wprowadzony ręcznie. Takiego rekordu NIE możemy używać w planowaniu, ponieważ nie zostanie od odesłany do Bazusa.

- **Nieaktywne rekordy.**



- Jeżeli rekord w BAZUS został zaznaczony jako nieaktywny, wówczas rekord staje się nieaktywny również w Plansoft.org.
- Jeżeli nieaktywny rekord w Plansoft.org został wcześniej zaimportowany z Bazus (to znaczy, nie jest to historyczny rekord), to dostęp do rekordu jest blokowany (rekord jest ukrywany).
- Nieaktywny rekord nie jest aktualizowany (imię, nazwisko itd. nie są już aktualizowane)
- Dane, których nie ma w Bazus (kolory, skróty) są generowane przez integrację automatycznie.
- Importowane dane słownikowe są widoczne dla wszystkich planistów i ról.
- Plan studiów.
  - Do Plansoft.org przesyłane są tylko poprawne kombinacje, to znaczy takie, gdzie wszystkie pola są wypełnione: wykładowca, grupa, przedmiot, forma zajęć, liczba zajęć.
  - W Plansoft.org musi zostać założony semestr o nazwie odpowiadającej dokładnie nazwie semestru założonej w systemie Bazus. W razie, gdy semestr nie zostanie odnaleziony, informacja zostaje zaraportowana w logu.
  - **Ważne!** Ponadto nazwa tego semestru musi zostać wpisana w polu Semestr na formularzu **Plik | Integracja**. Pobierane są tylko dane dla tego, wybranego semestru.

### Relacje pomiędzy grupami

Relacje pomiędzy grupami (nadrzędny, podrzędny) nie są importowane z BAZUS.

Relacje należy utworzyć ręcznie w Plansoft.org. Relacje pomiędzy grupami zbudowane po stronie plansoft.org nie są przenoszone do BAZUSa (nie są brane pod uwagę).

### Pobranie danych do Plansoft.org- harmonogram

Synchronizacja danych słownikowych uruchamiana jest automatycznie godzinę- nie nie musisz robić.

Plan studiów aktualizowany jest również automatycznie, raz dziennie, o godz. 04.00, ale może być wykonany o dowolnej porze poprzez naciśnięcie przycisku **Pobierz Plan**.

Naciśnięcie przycisku **Pobierz Plan** nie powoduje wysłania rozkładu natychmiast – w systemie rejestrowane jest zlecenie wysłania rozkładu, które zostanie zrealizowane o pełnej godzinie.

Import słowników trwa kilka sekund, import planu studiów trwa około dwóch minut i jest realizowany poza godzinami pracy z uwagi na konieczność przeliczenia liczby zaplanowanych zajęć dla całej bazy danych.

### Jak sprawdzić, czy przesyłanie danych działa?

W oknie **Plik | Integracja** sprawdzamy, czy były jakieś błędy w trakcie integracji oraz kiedy ostatni raz dane były przesyłane.



Nie musimy wczytywać się w znaczenie symboli na ekranie: ważne jest, tylko to, aby w kolumnie Uruchomiono była dzisiejsza data, a komunikat kończył się słowem „OK”.

Integracja

Podstawowe
Konfiguracja

Semestr

2022/2023Z

Pobierz Plan

Wyślij Rozkład

Raport: Co NIE zostanie wysłane do Bazusa?

Uruchomiono	Komunikat
2022-12-08 09:00:05	TO PLANSOFT_DCT:N: OK
2022-12-08 08:00:04	TO PLANSOFT_DCT:N: OK
2022-12-08 07:00:05	TO PLANSOFT_DCT:N: OK
2022-12-08 06:00:04	TO PLANSOFT_DCT:N: OK
2022-12-08 05:00:05	TO PLANSOFT_DCT:N: OK
2022-12-08 04:01:43	TO PLANSOFT_PLAN:N: OK
2022-12-08 04:00:05	TO PLANSOFT_DCT:N: OK
2022-12-08 03:00:05	TO PLANSOFT_DCT:N: OK

## Planujemy zajęcia!

No cóż, wreszcie możemy rozpocząć planowanie 😊

## Przesyłamy rozkład do BAZUS

Rozkład przesyłamy za pomocą dobrze już znanego formularza **Plik | Integracja**.

Przycisk **Wyślij Rozkład** naciskamy, gdy rozkład jest ukończony.

Przycisk **Wyślij Rozkład** powinien być naciskany, gdy nikt inny nie pracuje w systemie Plansoft.org. Rozkład możemy przysłać (aktualizować) wielokrotnie.

Warunkiem koniecznym do tego, aby zajęcia zostały wysłane do Bazusa jest to, że musi istnieć powiązany Plan Studiów, zaimportowany z Bazus (pole Integration Id nie puste).

Integracja

Podstawowe
Konfiguracja

Semestr

2019/2020L

Pobierz Plan

Wyślij Rozkład

Raport: Co NIE zostanie wysłane do Bazusa?

Naciśnięcie przycisku nie powoduje wysłania rozkładu natychmiast – w systemie rejestrowane jest zlecenie wysłania rozkładu, które zostanie zrealizowane o pełnej godzinie.



Integracja

Podstawowe Konfiguracja

Semestr  
2019/2020L

Pobierz Plan

Wyślij Rozkład

Raport: Co NIE zostanie wysłane do Bazusa?

Zlecenie wysłania rozkładu zajęć zostało przyjęte i zostanie wkrótce wykonane.

Przed wysłaniem, zajęcia są scalane (sąsiednie bloki są łączone).

### Raport: Co nie zostało wysłane do Bazusa?

Aby upewnić się, że wszystkie zajęcia zostały wysłane do Bazusa, naciskamy przycisk **Raport: Co nie zostanie wysłane do Bazusa?** Generowanie raportu trwa kilka minut.

Raport pokazuje zajęcia, który NIE można przesłać do Bazusa, gdy raport nie pokaże żadnych rekordów - to dobrze!

Przykładowy raport:

Zajęcia bez planu studiów

Id	Dzień	Godz. od	Godz. do	Dydaktyk	Grupa	Forma	Przedmiot	Sala
1270161	2022-09-23	11:30	13:00	4000925 dr ANNA SZYMCZAK 34	4005002 2021Z_ES1_FR_W1	4001800 Egz.	4001099 Makroekonomia 154	4001229 s. 103 Sopot - Rzemieślnicza
1251710	2022-09-23	17:00	18:30	4013504 mgr ALICJA KARAŚ 695	4065342 2021Z_AW1_AU_C1; NST_I ST_ARCHITEKTURA 678	4014771 Eduportal Ćwiczenia	4001078 Język angielski 108	
1250153	2022-09-23	18:45	20:15	4017018 mgr EWA ZACHARSKA-WOJTASZCZYK 36	4005700 2021Z_AN1_ZP_W1 674	4014771 Eduportal Ćwiczenia	4001078 Język angielski 108	

### Dla informatyków: szczegóły techniczne

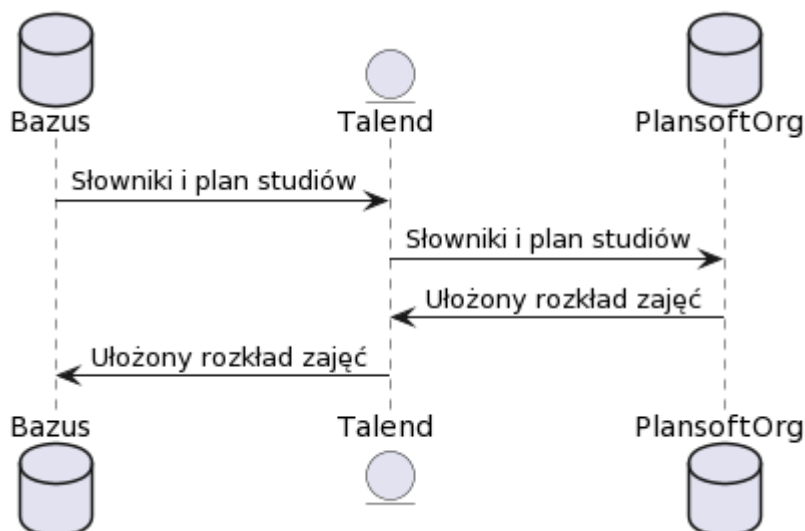
#### Jak to zostało zrobione?

Integracja składa się z trzech części:

1. Przesłanie danych słownikowych,
2. Przesłanie planu studiów,
3. Odesłanie ułożonego rozkładu zajęć.

Program napisano w technologii Talend Data Integration Studio.

Dane są przesyłane pomiędzy bazami danych Oracle oraz MSSQL.



<https://www.planttext.com/>

```

@startuml
database Bazuś
entity Talend
database PlansoftOrg
  
```

```

Bazuś -> Talend : Słowniki i plan studiów
Talend -> PlansoftOrg : Słowniki i plan studiów
PlansoftOrg -> Talend : Ułożony rozkład zajęć
Talend -> Bazuś : Ułożony rozkład zajęć
@enduml
  
```

### Przesłanie danych słownikowych

**Kiedy:** Co godzinę.

Przygotowanie danych w MSSQL	Procedura osadzana: IntPZ_zestaw_dydaktykow_v1 IntPZ_zestaw_grup_v1 IntPZ_zestaw_sal_v1 IntPZ_zestaw_przedmiotow_v1 IntPZ_zestaw_form_zajec_v1
Przesłanie danych MSSQL Odczyt -> Oracle Zapis	bazus.dbo.IntPZ_zestaw_dydaktykow -> int_lecturers bazus.dbo.IntPZ_zestaw_grup -> int_groups bazus.dbo.IntPZ_zestaw_sal -> int_resources bazus.dbo.IntPZ_zestaw_przedmiotow -> int_subjects bazus.dbo.IntPZ_zestaw_form_zajec -> int_forms
Przetwarzanie Oracle	Procedura osadzana: begin integration.int_to_plansoft_dict(); end;

### Przesłanie planu studiów

**Kiedy:** Raz na dobę wieczorem oraz na żądanie użytkownika o pełnych godzinach.

Oracle: Sprawdzenie, czy interfejs powinien zostać uruchomiony.	<pre> select Value from system_parameters where name = 'RUN_INT_TO_PLANSOFT_PLAN' and value='YES'           </pre>
---	--



Przetwarzanie MSSQL	Procedura osadzana: IntPZ_plan_studiow_v1
Przesłanie danych MSSQL -> Oracle	bazus.dbo.IntPZ_plan_studiow->int_plan
Przetwarzenie Oracle	Procedura osadzana: begin integration.int_to_plansoft_plan('N'); end;
Oracle Potwierdzenie zakończenia przetwarzania	begin delete from system_parameters where name = 'RUN_INT_TO_PLANSOFT_PLAN'; commit; end;

### Odesłanie ułożonego rozkładu zajęć

**Kiedy:** Raz na dobę wieczorem oraz na żądanie użytkownika o pełnych godzinach.

Oracle: Sprawdzenie, czy interfejs powinien zostać uruchomiony.	begin delete from system_parameters where name = 'RUN_INT_TO_PLANSOFT_PLAN'; commit; end
Przetwarzenie Oracle	begin integration.int_from_plansoft(); end;  Zajęcia są scalane (sąsiednie bloki są scalane)
Przesłanie danych Oracle -> MSSQL	int_classes->IntPZ_plan_zajec_event int_class_members-> IntPZ_plan_zajec_event_member
Przetwarzanie MSSQL	Proedura osadzana IntPZ_planner_to_bazus
Oracle Potwierdzenie zakończenia przetwarzania	begin delete from system_parameters where name = 'RUN_INT_FROM_PLANSOFT'; commit; end

### Bazus\_sub\_map

Bazus przekazuje przedmioty wymnożone przez formy prowadzenia zajęć, np. przedmiot matematyka przesyłany jest trzykrotnie, jeżeli prowadzonych jest w ramach trzech form zajęć. Aby uniknąć wyświetlania wielokrotnie tego samego przedmiotu, do plansoft.org przedmiot jest importowany tylko jeden raz. W tym celu używamy tabeli mapującej bazus\_sub\_map: wiele przedmiotów w Bazus –jeden przedmiot w plansoft.org.

Najlepiej wyjaśnić mechanizm działania tabeli na przykładzie:

Bazus wysyła trzykrotnie ten sam przedmiot z trzema różnymi integration\_id:

Integration_id	Przedmiot	Forma
1	Matematyka	Ćwiczenia



2	Matematyka	Wykład
3	Matematyka	Egzamin

Do plansoft.org importujemy tylko jeden rekord (a nie trzy rekordy)

Integration_id	Przedmiot
1	Matematyka

W tabeli bazus\_sub\_map tworzymy mapowanie:

Integration_id	Przedmiot	Forma	Plansoft. Przedmiot ID
1	Matematyka	Ćwiczenia	1
2	Matematyka	Wykład	1
3	Matematyka	Egzamin	1

Mapowanie używane jest kilka razy:

1. Gdy zasilamy słownik Przedmioty
2. Gdy zasilamy słownik Plan Studiów
3. Gdy odsyłamy gotowy rozkład do Bazusa.

W plansoft.org planista planuje zajęcia:

Kiedy	Przedmiot	Forma
2022.12.20 godz.8.00	Matematyka (Id=1)	Wykład
2022.12.20 godz.8.00	Matematyka (Id=1)	Ćwiczenia

Do bazusa wysyłamy zajęcia:

Kiedy	Przedmiot	Forma
2022.12.20 godz.8.00	Matematyka (integration Id=2)	Wykład
2022.12.20 godz.8.00	Matematyka (integration Id=1)	Ćwiczenia

### Zastosowany Oprogramowanie

Talend Data Integration Studio

Java AZUL

SQLDeveloper (ORACLE)

SSMS (Microsoft SQL Server Management Studio)

Notepad++

### Instalacja połączenia

1. Przed pierwszym uruchomieniem interfejsu uzupełniliśmy w plansoft.org integration\_id, miało to na celu uniknięcie utworzenia duplikatów rekordów po stronie plansoft.org. Zastosowano następujące reguły:
  - a. Weryfikujemy, czy rzeczywiście integration\_id generowany przez system źródłowy jest unikatowy.





b. Porównujemy i uzupełniamy dane:

```
Wykładowcy istniejący w Bazus i nie istniejący w plansoft.org
select * from
(
select title, first_name, last_name from int_lecturers where is_active=1
minus
select title, upper(first_name), upper(last_name) from lecturers
) order by 3,2

Wykładowcy istniejący w plansoft.org i nie istniejący w Bazus
select * from
(
select title, upper(first_name), upper(last_name) from lecturers where id >0
minus
select title, first_name, last_name from int_lecturers where is_active=1
) order by 3,2

-----
-- LEC
update lecturers l set integration_id = (select integration_id from
int_lecturers where upper(first_name)=upper(l.first_name) and
upper(last_name)=upper(l.last_name) and is_active=1 );

update lecturers l set title = (select title from int_lecturers where
upper(first_name)=upper(l.first_name) and upper(last_name)=upper(l.last_name)
and is_active=1 )
where (select title from int_lecturers where
upper(first_name)=upper(l.first_name) and upper(last_name)=upper(l.last_name)
and is_active=1 ) is not null

--ROM
update rooms l set integration_id = (select integration_id from int_resources
where name=l.name and is_active=1 ); --and location=l.attribs_01

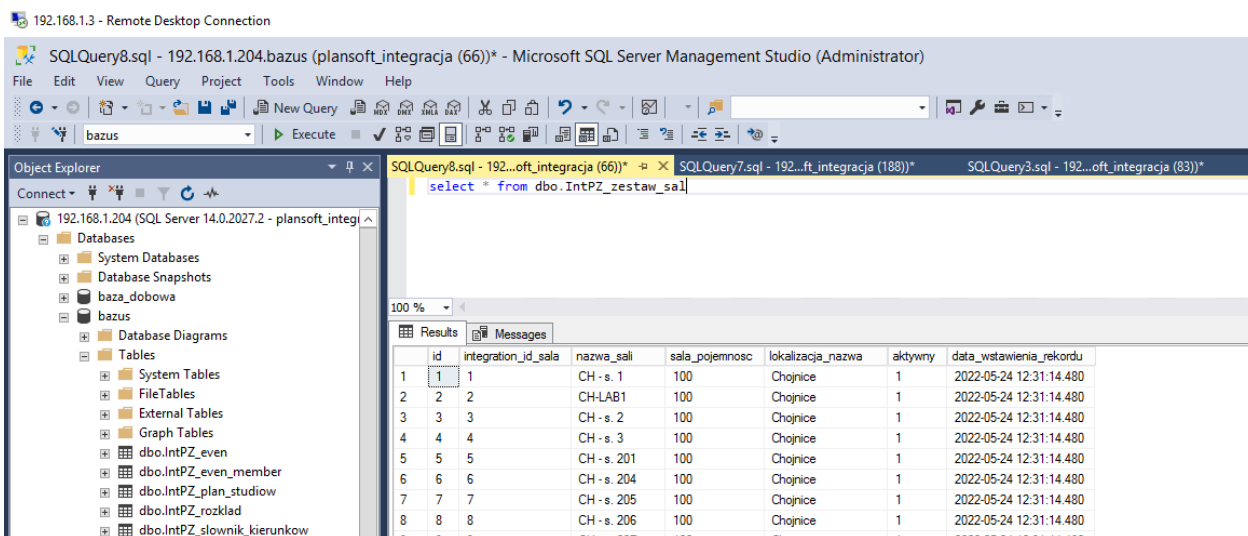
--GRO
update groups l set integration_id = (select integration_id from int_groups
where name=l.abbreviation ) and integration_id is null;

--FOR
update forms set Name='(Nieaktywne)Praktyka', abbreviation='(X)Praktyka',
is_active='0' where Id=4007360;
update forms set name = initcap(name);
update forms m set integration_id = (select integration_id from int_forms where
name = m.name );
commit;

--GRO
update groups g set integration_id = (select integration_id from int_groups
where upper(name)=upper(l.first_name) and upper(last_name)=upper(l.last_name)
and is_active=1);

--SUB
Ze względu na to, że Bazus zwraca kombinację: Przedmiot + forma, matchowanie nie było
wykonywane
```

- Na serwerze integracyjnym instalujemy *Microsoft SQL Server Management Studio*, konfigurujemy połączenie i upewniamy się, że możemy odczytać dane z tabel integracyjnych.



Upewniamy się, że poniższe zapytania SQL działają

```
select * from bazus.dbo.IntPZ_zestaw_dydaktykow
select * from bazus.dbo.IntPZ_zestaw_grup
select * from bazus.dbo.IntPZ_zestaw_sal
select * from bazus.dbo.IntPZ_zestaw_przedmiotow
select * from bazus.dbo.IntPZ_zestaw_form_zajec
select * from bazus.dbo.IntPZ_plan_studiow
```

3. Aktywujemy integrację i konfigurujemy parametry integracji.

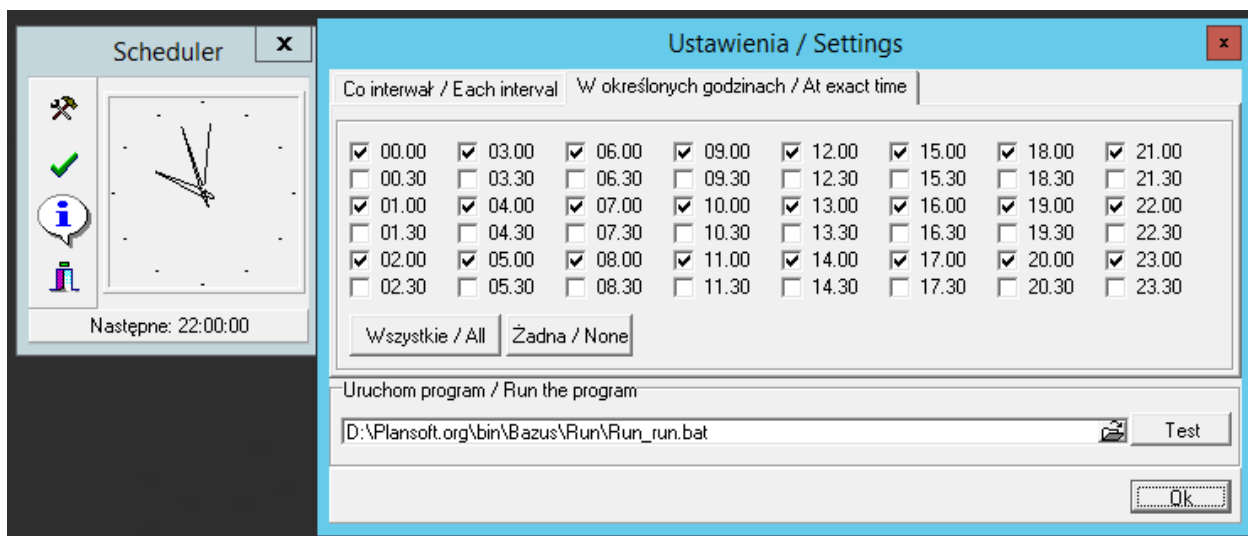
1.

```
insert into system_parameters (name, value) values ('INT_IS_ACTIVE', '1');
Commit;
```

2. Za pomocą formularza **Plik | Integracja** wprowadzamy Semestr i typ ograniczenia.

W polu **Typ ograniczenia** wybieramy: Semestr, Wykładowca, Przedmiot, Forma, Grupa.

4. Na serwerze harmonogramujemy usługę napisaną w Talend Data Integration Studio, która przesyła dane pomiędzy Bazusem (MSSQL) a Plansoft.org (Oracle).



D:\Plansoft.org\bin\Bazus\Run\Run\_run.bat

### 5. Harmonogramujemy proces synchronizacji planu zajęć.

```

Begin
  dbms_scheduler.create_job(
    job_name => 'INT_BAZUS'
    ,job_type => 'PLSQL_BLOCK'
    ,job_action => 'begin insert into system_parameters (name, value)
values(''RUN_INT_TO_PLANSOFT_PLAN'', ''YES''); commit; end;'
    ,repeat_interval => 'freq=daily; byhour=3'
    --,repeat_interval => 'freq=minutely'
    ,enabled => TRUE
    ,comments => '');
--DISPLAY SCHEDULED JOBS: select * from dba_scheduler_jobs
--DROP JOB                : begin dbms_scheduler.drop_job('INT_BAZUS'); end;
--CLEAR LOG               : delete from xxmsztools_eventlog where module_name = 'INT_TO_PLANSOFT';
--DISPLAY LOGS            : select * from xxmsztools_eventlog where module_name = 'INT_TO_PLANSOFT' order by id desc
end;

```

## Rozwiązywanie problemów

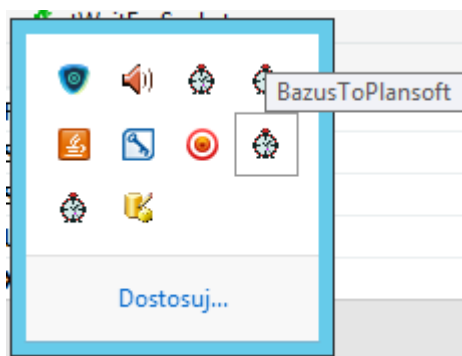
1. Za pomocą **Plansoft.org | Plik | Integracja** sprawdź logi- czy są świeże wpisy w logach? Jeżeli podczas synchronizacji pojawia się błąd, to jest on zapisywany w logu synchronizacji. Logi możesz również sprawdzić za pomocą SQL:

```

select * from xxmsztools_eventlog where module_name = 'INT_TO_PLANSOFT'
order by id desc

```

2. Zaloguj się do serwera 192.168.1.3. Sprawdź, czy zadanie jest uruchomione na serwerze. Jeżeli nie, to uruchom je



3. Zaloguj się do serwera 192.168.1.3. Uruchom Talend Data Integration Studio. Uruchom zadania ręcznie w celu przesledzenia, czy występują błędy.

### Po co jest parametr pCleanpMode w pakiecie Integration?

Parametr kasuje rekordy przed ich utworzeniem – w normalnym trybie pracy nie trzeba tego robić.

Parametr może być pomocny podczas rozwiązywania problemów.

Procedura	Akcja gdy pCleanpMode = true
<code>procedure int_to_plansoft_dict (pCleanYpMode varchar2 default 'N');</code>	<code>Truncate table LEC_PLA, GRO_PLA, ROM_PLA, SUB_PLA in FOR_PLA</code>
<code>procedure int_to_plansoft_plan (pCleanYpMode varchar2 default 'N');</code>	<code>delete from TT_COMBINATIONS where integration_id is not null delete from tt_resource_lists where tt_comb_id in (select Id from TT_COMBINATIONS where integration_id in (select integration_id from TT_INTERFACE ) ); delete from TT_INCLUSIONS where tt_comb_id in (select Id from TT_COMBINATIONS where integration_id in (select integration_id from TT_INTERFACE ) ); delete from tt_cla where tt_comb_id in (select id from tt_combinations where weight = 122 and per_id = nvl(pPER_ID, per_id) and integration_id is not null);</code>
<code>procedure int_from_plansoft (pCleanYpMode varchar2 default 'N');</code>	<code>delete from tt_cla where tt_comb_id in (select id from tt_combinations where weight = 122 and per_id = nvl(pPER_ID, per_id) and integration_id is not null);</code>

### ORA-00001: unique constraint (PLANNER.ROOM\_UK) violated

```
drop index ROOM_UK;

merge into rooms m using
(
  select integration_id
    , name
    , capacity
    , location
    , orguni_id
    , is_active
  from int_resources
 where is_active=1
) int
on (m.integration_id = int.integration_id)
  when not matched then insert (
    id,
```



```

        name,
        rescat_id,
        attribn_01,
        attribs_01,
        integration_id,
        orguni_id,
        colour,
        is_active
    )
values (
    main_seq.nextval
,   int.name
,   1
,   int.capacity
,   int.location
,   int.integration_id
,   nvl(int.orguni_id,111)
,   round(dbms_random.value(128,255)) +
256*round(dbms_random.value(128,255)) +
256*256*round(dbms_random.value(128,255))
,   int.is_active
)
when matched then update
set name=int.name
,   attribn_01=int.capacity
,   attribs_01=int.location
,   orguni_id=nvl(nvl(int.orguni_id,orguni_id),111)
,   is_active = int.is_active;

--records in error
select id, integration_id, name, attribs_01 from rooms where (name, attribs_01)
in (select name, ATTRIBS_01 from rooms group by name, ATTRIBS_01 having
count(1)>1) order by name, attribs_01, integration_id;

--repeat the cleanup for each pair until all records are fixed
declare
affected_integration_id number := 52;
id_delete number := 4320300; --has NOT NULL integration_id
id_remain number := 4012881; --leave old (existing record)
begin
update rooms set integration_id = -affected_integration_id where id=id_delete;
update rooms set integration_id = affected_integration_id where id=id_remain;
update rom_cla set rom_id=id_remain where rom_id= id_delete;
delete from rooms where id=id_delete;
commit;
end;

CREATE UNIQUE INDEX "PLANNER"."ROOM_UK" ON "PLANNER"."ROOMS" (CASE "RESCAT_ID"
WHEN 1 THEN "NAME"||' '||"ATTRIBS_01" ELSE TO_CHAR("ID") END )
TABLESPACE "USERS" ;

```

**ORA-00001: unique constraint (PLANNER.SUB\_ABBREVIATION\_I) violated**

```
drop index SUB_ABBREVIATION_I;
```

```
begin integration.int_to_plansoft_dict(); end;
```

```
--records in error
```

```
select id, integration_id, ABBREVIATION, name from subjects where (ABBREVIATION)
in (select ABBREVIATION from subjects group by ABBREVIATION having count(1)>1)
order by ABBREVIATION, integration_id;
```

Scal rekordy za pomoca standardowej funkcjonalnosci plansoft.org

Lub uzyj tego skryptu:

```
begin
```

```
--reset abbr. I will be properly set by the interface
```

```
update subjects set abbreviation='4325370' where id = 4325370;
```

```
update subjects set abbreviation='4046179' where id = 4046179;
```

```
update subjects set abbreviation='4325384' where id = 4325384;
```

```
update subjects set abbreviation='4046330' where id = 4046330;
```

```
update subjects set abbreviation='4325378' where id = 4325378;
```

```
update subjects set abbreviation='4046382' where id = 4046382;
```

```
commit;
```

```
end;
```

```
CREATE UNIQUE INDEX "PLANNER"."SUB_ABBREVIATION_I" ON "PLANNER"."SUBJECTS"
("ABBREVIATION")
TABLESPACE "USERS" ;
```

**ORA-00001: unique constraint (PLANNER.GRO\_ABBREVIATION\_I) violated**

```
drop index GRO_ABBREVIATION_I;
```

```
begin integration.int_to_plansoft_dict(); end;
```

```
select * from xxmsztools_eventlog where module_name = 'INT_TO_PLANSOFT' order by id desc
```

```
select Id, ABBREVIATION, name, integration_id, to_char(creation_date,'yyyy-mm-dd')
yyyymmdd, created_by, group_type from groups where ABBREVIATION in (select
ABBREVIATION from groups group by ABBREVIATION having count(1)>1)
```

Scal rekordy za pomoca standardowej funkcjonalnosci plansoft.org

```
CREATE UNIQUE INDEX "PLANNER"."GRO_ABBREVIATION_I" ON
"PLANNER"."GROUPS" ("ABBREVIATION") TABLESPACE "USERS" ;
```

**Przedmioty zawierające znaki końca wiersza w nazwach**

```
select id, name, integration_id from subjects where ascii(substr(name,length(name),1))=10
update subjects set name = replace(name,chr(10),")
```

**ORA-00001: unique constraint (PLANNER.SUB\_NAME\_UI) violated**

```
drop index SUB_NAME_I;
```

```
select id, abbreviation, name, integration_id from subjects where name in (select name from subjects  
group by name having count(1)>1) order by name
```

Scal przedmioty ręcznie

```
CREATE UNIQUE INDEX "PLANNER"."SUB_NAME_I" ON "PLANNER"."SUBJECTS"  
("NAME")
```

**ORA-00001: unique constraint (PLANNER.LEC\_NAME\_UI) violated**

```
drop index LEC_NAME_UI;
```

```
begin integration.int_to_plansoft_dict(); end;
```

```
select * from xxmsztools_eventlog where module_name = 'INT_TO_PLANSOFT' order by id desc
```

```
select Id, FIRST_NAME, LAST_NAME, TITLE, integration_id, to_char(creation_date,'yyyy-mm-  
dd') yyyymmdd, created_by
```

```
from LECTURERS where (FIRST_NAME, LAST_NAME, TITLE) in (select FIRST_NAME,  
LAST_NAME, TITLE from LECTURERS group by FIRST_NAME, LAST_NAME, TITLE having  
count(1)>1)
```

Scal rekordy za pomoca standardowej funkcjonalnosci plansoft.org

```
CREATE UNIQUE INDEX "PLANNER"."LEC_NAME_UI" ON "PLANNER"."LECTURERS"  
("FIRST_NAME", "LAST_NAME", "TITLE")  
TABLESPACE "USERS" ;
```

**Scalanie istniejącego przedmiotu z nowym przedmiotem**

```
update subjects set name = name || '(X)' where integration_id is null;  
select to_fix.*  
  , 'update classes set sub_id='||id_to_retain ||' where sub_id='||it_to_delete||';' sql  
from  
(select Id it_to_delete  
  , name  
  , integration_id  
  , (select id from subjects where name = m.name||'(X)' and integration_id is null ) id_to_retain  
from subjects m  
where integration_id is not null and name in (select replace(name,'(X)','') from subjects where integration_id is null and name like '%(X)%')  
) to_fix  
  
begin  
for rec in (  
  select to_fix.*  
  from  
  (select Id it_to_delete  
    , name  
    , integration_id  
    , (select id from subjects where name = m.name||'(X)' and integration_id is null ) id_to_retain  
  from subjects m  
  where integration_id is not null and name in (select replace(name,'(X)','') from subjects where integration_id is null and name like '%(X)%')  
) to_fix  
) loop  
  update subjects set integration_id = -rec.integration_id where id=rec.it_to_delete;  
  update subjects set integration_id = rec.integration_id where id=rec.id_to_retain;  
  delete from subjects where id=rec.it_to_delete;  
  commit;  
end loop;  
commit;  
end;
```



### Sprawdzenie, które pozycje z planu zajęć nie mogą być zaimportowane

```
select int_plan.*
, (select count(id) from lecturers where integration_id=integration_id_lec) lec_ok
, (select count(id) from groups where integration_id=integration_id_gro) gro_ok
, (select count(id) from subjects where integration_id=integration_id_sub) sub_ok
, (select count(id) from forms where integration_id=integration_id_for) for_ok
from int_plan where integration_id in (
select integration_id from int_plan
minus
select integration_id from tt_combinations
)
and cycle_name='2022/2023Z'
and is_active='1'
```