

ERP NOT-SAP Integration Specification

This document describes the requirements for integrating ERP-not SAP to NeXT Improve System.

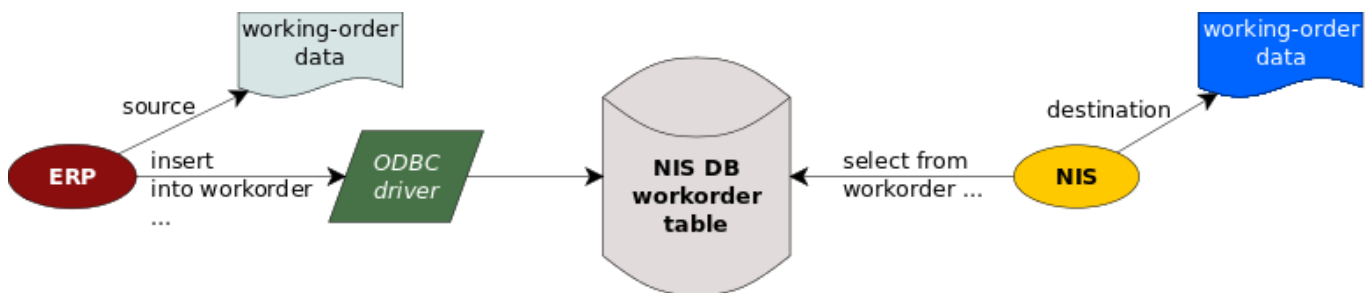
Definitions

Word	Definition
ERP	not-SAP-FOX ERP System used by the plant
NIS	NeXT Improve System for OEE Monitoring
ERP-Users	Plant's staff that uses ERP
ERP integration procedure	Procedure implemented into ERP by ERP Developers for integrating NIS
WO / work-order	Working-Order referred to one specific kind of product and related quantity to be produced
DBMS	Database Management System
NIS DB	NIS database that is a relational database using Postgres DBMS
ODBC Driver	Driver ODBC for accessing Postgres DBMS
SQL	Structured query language
ruek	working-order identifier used in general by SAP-FOX; if ERP doesn't use this concept, it will be not considered

Integration Criterions

The integration between NIS and ERP is implemented using some border tables defined into NIS DB.

ERP must be able to insert information required by NIS into NIS DB using standard SQL language. The access to NIS DB should be executed through ODBC Driver for Postgres. Following a general flow-chart



Working-order data are the source of information; first they are available into ERP database and are accessible by ERP System, but not by NIS. So ERP inserts information required by NIS into working-order border table; border table is provided into NIS DB.

After ERP inserts data into border table, NIS detects and saves them into destination table (operative table).

NeXT's technicians will supply connection parameters, that are following:

- **IP Address, Port:** Server DBMS IP address and port
- **User name, password:** User name and password for accessing DBMS
- **Database Name:** name of NIS DB

Conceptual Schema

NIS requires the following working-order's information:

- **WO identifier:** unique identifier of the working-order
- **product identifier and description:** code and description of the product referred by the WO
- **quantity to be produced:** number of items to be produced

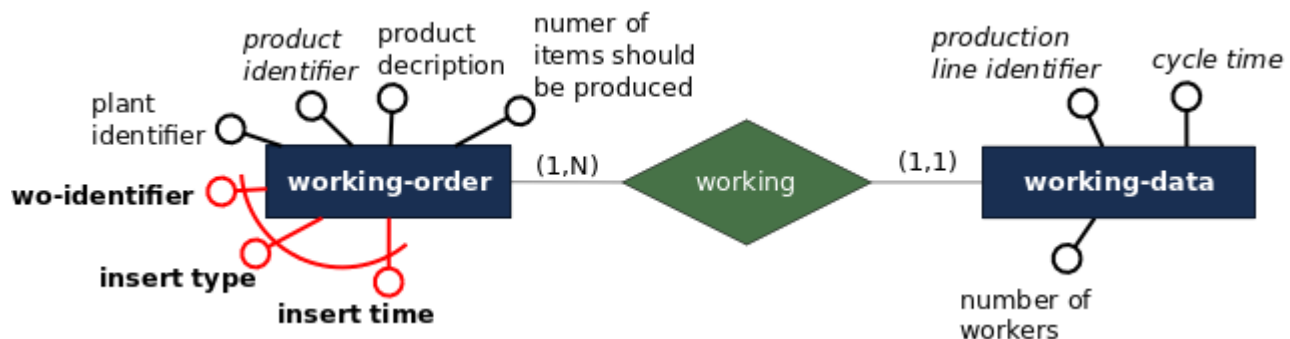
Each WO refers also to following production information:

- **production lines:** code of the production lines (assembly lines / machines) in which the WO should be produced
- **cycle time:** the period in seconds, required to complete one cycle of production (production of one item) for each production lines; it corresponds exactly to takt time in case of mechanical line
- **number of workers:** number of workers assigned to WO (significant only for assembly line for calculating takt time through cycle time)

Furthermore it's necessary to manage some flags that permit the identification of working-order's information status; following:

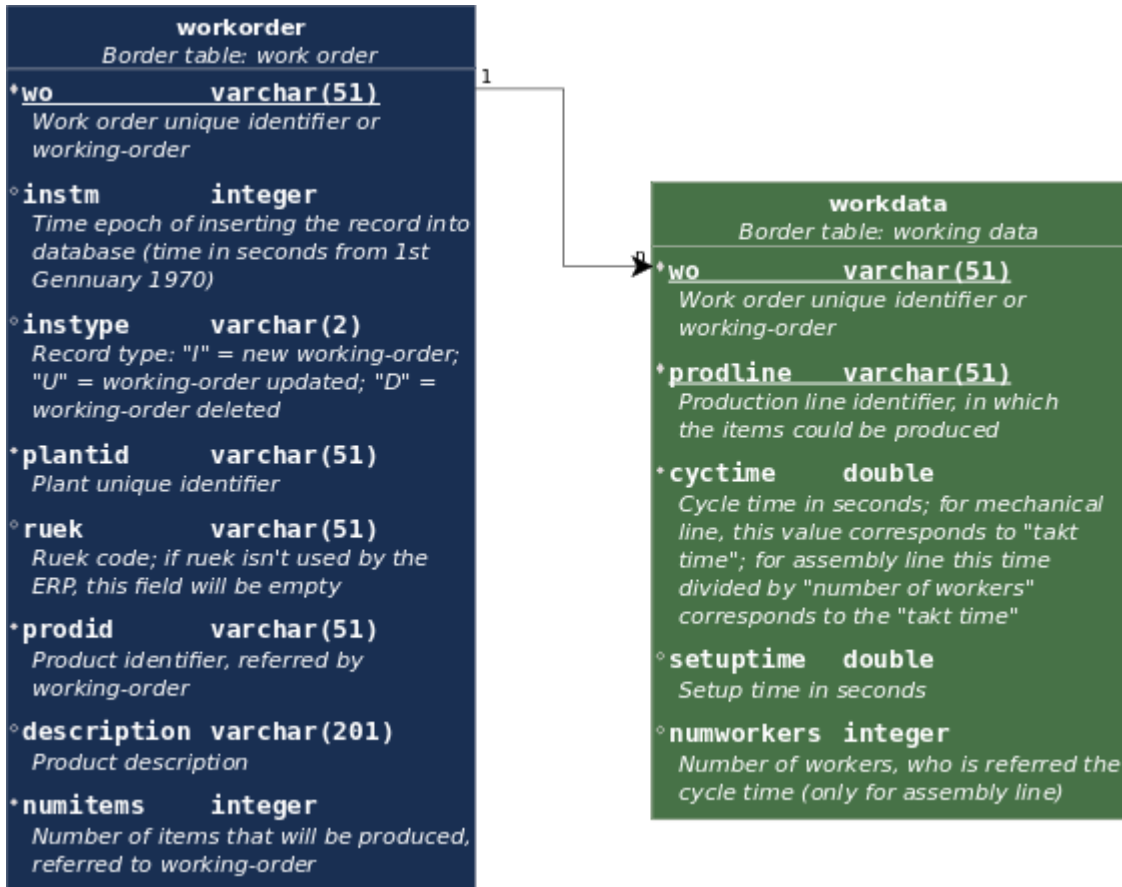
- **time status:** time in which the information is generated
- **information status:** status of the information, distinguishing as following:
 - *inserted information:* information to be inserted at first time
 - *updated information:* information to be updated
 - *deleted information:* information to be deleted

Following a conceptual schema:



Logical Schema

NIS DB will provide border tables for ERP integration; two border table are required for implementing integration. Following logical schema of the border tables



The name of the first border table is "workorder", following fields list:

Field Name	Field Type	Field Description	Primary key	Nullable
wo	varchar(51)	Unique identifier of working-order	Yes	No
instm	integer	Time epoch of inserting the record into database (time in seconds from 1st Gennuary 1970)	Yes	No
instype	varchar(2)	Record type: "I" = new working-order; "U" = working-order updated; "D" = working-order deleted	Yes	No
plantid	varchar(51)	Plant unique identifier		
ruek	varchar(51)	Ruek code; if ruek isn't used by the ERP, this field will be empty	No	Yes
prodid	varchar(51)	Product identifier, referred by working-order	No	No
description	varchar(201)	Product description	No	Yes
numitems	integer	Number of items that will be produced, referred to working-order	No	No

The primary key is defined on the following fields:

- **<wo, instm, instype>**

It implies that a specific WO (wo) at a specific time (instm) for a specific record type (instype), is related to a specific row of the table workorder.

The name of second shared table is "workdata", following fields list:

Field Name	Field Type	Field Description	Primary key	Nullable
wo	varchar(51)	Unique identifier of working-order	Yes	No

Field Name	Field Type	Field Description	Primary key	Nullable
prodline	varchar(51)	Production line identifier, in which the items could be produced	Yes	No
cycletime	double	Cycle time in seconds; for mechanical line, this value corresponds to "takt time"; for assembly line this time divided by "number of workers" corresponds to the "takt time"	No	No
setuptime	double	Setup time in seconds	No	No
numworkers	integer	Number of workers, who is referred the cycle time (only for assembly line); if ERP doesn't know number of worker this field should be empty	No	Yes

The primary key is defined on the following fields:

- **<wo, prodline>**

A foreign key is defined on field **wo** (it refers to field wo of table workorder). The relation between table **workorder** and table **workdata** is 1 to N respectively (as we can see into conceptual schema and also into logical schema). Therefore one working-order stored into **workorder** table corresponds to some rows into **workdata** table, one row for each production line in which working-order could be produced. This relation permits to deduce takt-time for each WO for each production line where WO could be produced.

Following an example of create table query for creating workorder/workdata table using standard SQL language

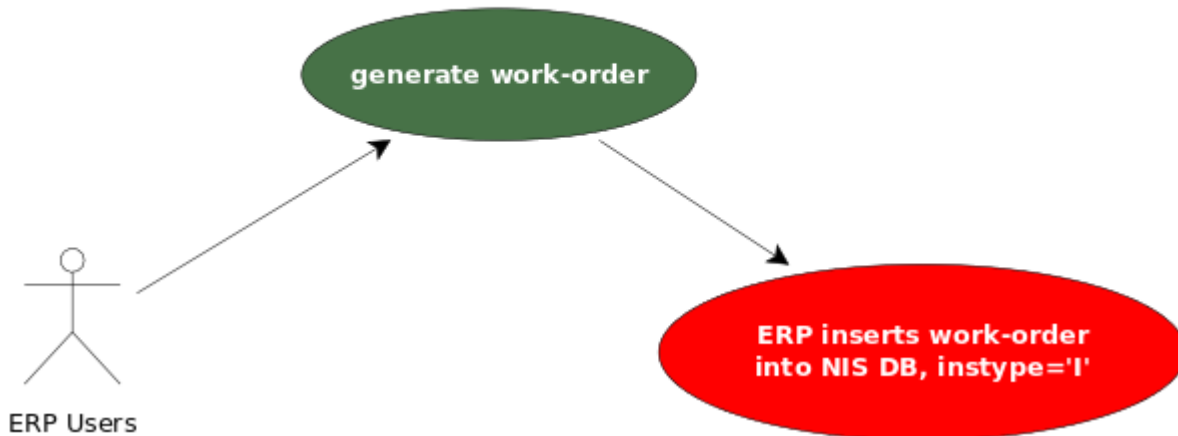
```
--working-order table and related constraint
CREATE TABLE workorder (
  wo VARCHAR(51) NOT NULL,
  instm INTEGER NOT NULL,
  instype VARCHAR(2) NOT NULL,
  plantid VARCHAR(51) NOT NULL,
  ruek VARCHAR(51),
  prodid VARCHAR(51) NOT NULL,
  description VARCHAR(201),
  numitems INTEGER NOT NULL,
  prodline VARCHAR(51) NOT NULL,
  cycletime FLOAT NOT NULL,
  numworkers INTEGER
);
ALTER TABLE workorder
  ADD CONSTRAINT ix_workorder_key PRIMARY KEY (wo, instm, instype);
--working-data table and related constraint
CREATE TABLE workdata (
  wo CHARACTER VARYING(51) NOT NULL,
  prodline CHARACTER VARYING(51) NOT NULL,
  cycletime DOUBLE PRECISION NOT NULL,
  setuptime DOUBLE PRECISION NOT NULL DEFAULT 0,
  numworkers INTEGER
);
ALTER TABLE workdata
  ADD CONSTRAINT ix_workdata_key PRIMARY KEY (wo, prodline);
```

Integration Scenarios

There are three integration scenarios:

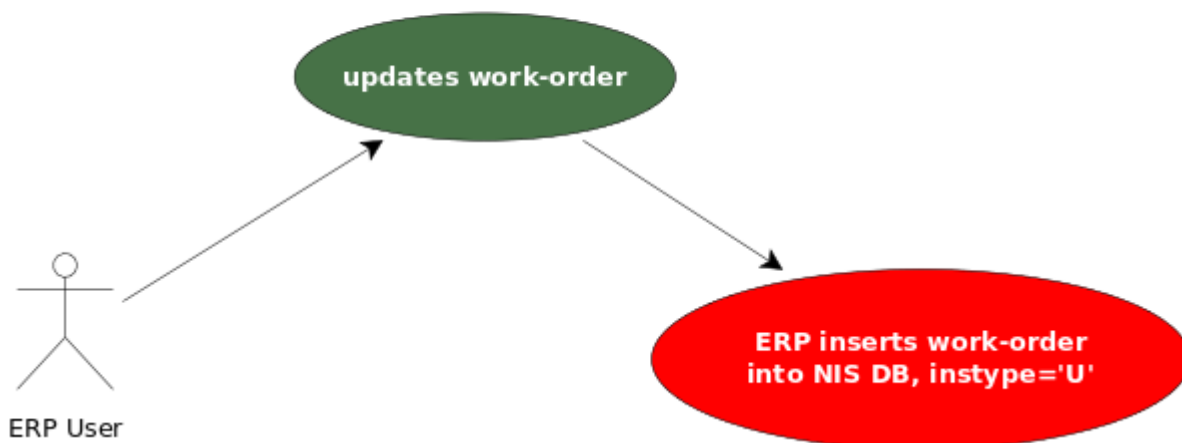
1. ERP-User inserts a new working-order into ERP
2. ERP-User updates an existing working-order into ERP
3. ERP-User deletes an existing working-order into ERP

USER inserts a new WO into ERP



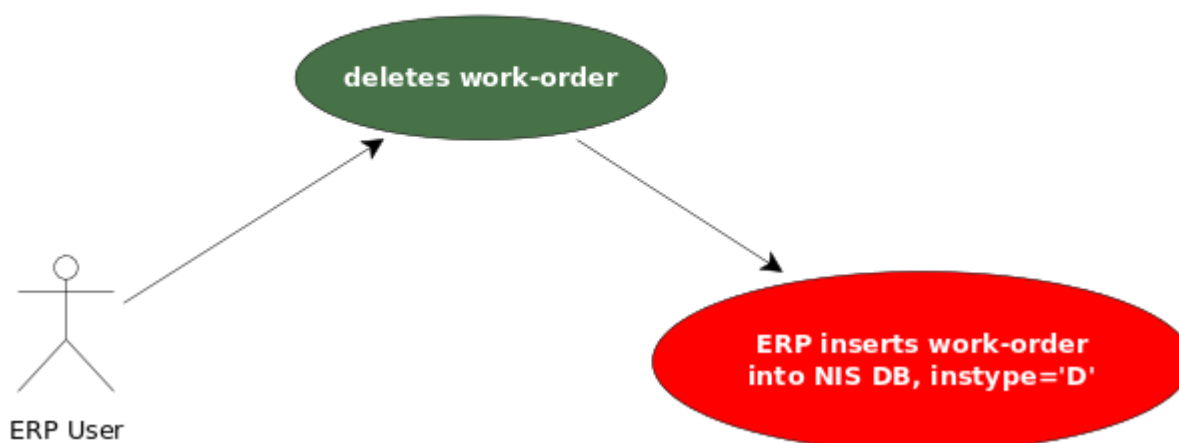
If an **ERP-User** inserts a new working-order into ERP, then ERP should insert data required into workorder/workdata tables (NIS DB); in this case the field **instm** should be set to the time of inserting record into NIS DB, while **instype** should be set to **I** ([I]nsert).

USER updates an existing WO into ERP



If an **ERP-User** updates an existing working-order into ERP, then ERP should insert data required into workorder/workdata tables (NIS DB); **before inserting data into workdata table the ERP should delete the workdata's rows related to the updated WO**; in this case the field **instm** should be set to the time of inserting record into NIS DB, while **instype** should be set to **U** ([U]pdate).

USER deletes an existing WO into ERP



If an **ERP-User** deletes an existing working-order into ERP, then ERP should insert data required into workorder table (NIS DB); in this case the field **instm** should be set to the time of inserting record into NIS DB, while **instype** should be set to **D** ([D]elete).

Integration Examples

We show an example in which ERP-User executes following steps:

1. first creates a new working-order identified by 44621852
2. then updates working-order 44621852 for changing number of items
3. finally deletes working-order 44621852

Naturally updating and deleting should be optionally operations and ERP-User should execute them based on his needs.

We begin considering that ERP-User inserts into ERP the following working-order

WO	PRODUCT ID	QTY(UNIT)
44621852	110.0017.976	20

This working-order could be produced on production line FAFL1 and FAFL2, with the following information

LINE	CYC TIME(sec)	NUM WORKERS
FAFL1	900	5
FAFL2	900	4

ERP integration procedure should provide to insert data into related fields of the table workorder / workdata; following relations between data in table above and fields in workorder / workdata tables

workorder		
Field Name	Data	Annotation
wo	44621852	
instime	1480712220	Corresponds to 2016-12-02 21:57
instype	I	
plantid	FFI	

workorder			
Field Name	Data	Annotation	
ruek	null		
prodid	110.0017.976		
description	COOKER HOOD XXXX	ERP extracts product description from internal table	
numitems	20		

workdata			
Num row	Field Name	Data	Annotation
1	wo	44621852	
1	prodline	FAFL1	Improve should know this code
1	cycletime	900	
1	numworkers	5	
2	wo	44621852	
2	prodline	FAFL2	Improve should know this code
2	cycletime	900	
2	numworkers	4	

Following query in SQL language that the ERP should execute on NIS DB

```
INSERT INTO workorder (wo, instm, instype, plantid, prodid, description, numitems)
VALUES ('44621852', 1480712220, 'I', 'FFI', '110.0017.976', 'COOKER HOOD XXXX',
20);
INSERT INTO workdata (wo, prodline, cycletime, numworkers)
VALUES ('44621852', 'FAFL1', 900, 5);
INSERT INTO workdata (wo, prodline, cycletime, numworkers)
VALUES ('44621852', 'FAFL2', 900, 4);
```

For reasons that we don't know, ERP-User updates the number of items from 20 to 30 in the same working-order; after that ERP updates working-order information as following:

WO	PRODUCT ID	QTY(UNIT)
44621852	110.0017.976	30

ERP integration procedure should provide to insert into NIS DB the change occurred. To do this ERP inserts into table workorder of NIS DB following data:

Field Name	Data	Annotation
wo	44621852	
instime	1480718520	Corresponds to 2016-12-02 23:42
instype	M	
plantid	FFI	
ruek	null	
prodid	110.0017.976	
description	COOKER HOOD XXXX	ERP extracts product description from internal table
numitems	30	

The query in SQL language that the ERP should execute on NIS DB is defined below:

```
DELETE FROM workorder WHERE wo='44621852';
DELETE FROM workdata WHERE wo='44621852';
INSERT INTO workorder (wo, instm, instype, plantid, prodid, description, prodline,
numitems, cycletime, numworkers)
VALUES ('44621852', 1480718520, 'M', 'FFI', '110.0017.976', 'COOKER HOOD XXXX',
```

```
'FAFL1', 30, 900, 5);
INSERT INTO workdata (wo, prodline, cycletime, numworkers)
VALUES ('44621852', 'FAFL1', 900, 5);
INSERT INTO workdata (wo, prodline, cycletime, numworkers)
VALUES ('44621852', 'FAFL2', 900, 4);
```

Not enough satisfied ERP-User adjudicates to delete the working-order from ERP system. ERP should communicate this event to NIS, so ERP inserts into workorder table of NIS DB following data:

Field Name	Data	Annotation
wo	44621852	
instime	1480719360	Corresponds to 2016-12-02 23:56
instype	D	
plantid	FFI	
ruek	null	
prodid	110.0017.976	
description	COOKER HOOD XXXX	ERP extracts product description from internal table
numitems	30	

ERP executes query SQL below on NIS DB:

```
DELETE FROM workorder WHERE wo='44621852';
DELETE FROM workdata WHERE wo='44621852';
INSERT INTO workorder (wo, instm, instype, plantid, prodid, description, prodline,
numitems, cycletime, numworkers)
VALUES ('44621852', 1480719360, 'D', 'FFI', '110.0017.976', 'COOKER HOOD XXXX',
'FAFL1', 30, 900, 5);
```

Additional Fields and Tables

This paragraph describes some additional fields and related types, usage, aims useful for some shared tables generalizations.

Sharde Table: workorder				
Field Name	Field Type	Field Size	Description	Optional (Y/N)
tmrel	integer	default	Time Epoch (time in seconds from 1 Jan 1970) referred to the time of releasing related work-order; it is used by NIS for sorting work-orders with the aim of providing the sequence of production	Y
tmclo	integer	default	Time Epoch (time in seconds from 1 Jan 1970) referred to the time of closing related work-order; it is never used by NIS	Y

Following queries to update database structure

```
--Time of releasing work-order
ALTER TABLE workorder ADD COLUMN tmrel INTEGER;
--Time of closing work-order
ALTER TABLE workorder ADD COLUMN tmclo INTEGER;
```

Table with the last import tm

```
CREATE TABLE wolasttm (
```



```
    lasttm INTEGER
);
COMMENT ON COLUMN wolasttm.lasttm IS 'Tm ultima importazione workorder';
```