

Udskrevet er dokumentet ikke dokumentstyret.

**Packagedesign**

| | | | | | |
|-----------------------------------------|-------------------------------------------|-----------------------------------------|---------------------------------------|----------------------------------------------------------|---------|
| Dokumentbrugere: KONC-POEM-MT | Dokumentansvarlig: D-PØMQansvar | Redaktør: Version: 1.2 | Dokumentnummer: 12. 6. 1.02 | Godkendt af: PAMA2610 17.08.2011 | Niveau: |
|-----------------------------------------|-------------------------------------------|-----------------------------------------|---------------------------------------|----------------------------------------------------------|---------|

- 1) General design principles
 - 1.1) Storage
 - 1.2) Packagenames
 - 1.3) Packagelogging
 - 1.4) The structure of a package
 - 1.5) Naming tasks
- 2) ODS-packages
- 3) EDW-packages

1) General design principles**1.1) Storage**

When packages are under development they are stored on the P: drive under konc-poem\70_BI\Projects. There is one directory per IS project. An IS project normally contains several IS packages.

When packages are deployed to test and prod, they should be placed in the relevant folder. The folders for deployment are:

\\gbdfs1\dsbdfs\Økonomi\KONC-POEM\70_BI\Deployed_packages\Prod
 \\gbdfs1\dsbdfs\Økonomi\KONC-POEM\70_BI\Deployed_packages\Test

Projects should be named with one of the following prefixes

| Prefix | Description |
|--------|-------------------------------------------------------------------------|
| ETL_ | Contains IS-packages that performs ETL |
| AS_ | Contains an Analysis services project |
| RS_ | Contains a Reportings services project |
| LEV_ | Contains ETL-processes that delivers data to customers (e.g. GD-tables) |

1.2) Packagenames

The following naming convention must be followed when a package is created, in order to organise the packages and understand its content.

Packages name syntax:

PRE_TRID_SHORTDESC

Packages syntax description:

| Syntax code | Description | Example |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| PRE | Prefix of the scheme where the output tables are stored | ods, edw, dm, etl |
| TRID | An ID to describe the main purpose of the package. For a full list of the valid ID's refer to the table below "Package purpose ID's". If more than one of these tasks are performed in the package, the criteria of the developer decides which is the main purpose. | DL, TR1 |
| SHORTDESC | A short description of the package main purpose, for example, If the package is a dataloader, it refers to a short description of the source system. If the package executes an EDW transformation, it refers to a short description of the table or group of tables refreshed in the package, etc. | LTD2 for Ltd2 source system dataloader, Togpersonale for a package refreshing Togpersonale table, etc |

Package purpose ID's:

| Code | Description |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DL_ | Data loader from source systems to ODS schema |
| TR1_, TR2_, etc | Transformations performed to the data in the EDW tables, such as, data checking, data cleaning, calculations or joins with other tables to add new data, etc. If more than one data transformations are performed to the same set of data, a sequential number is added to the prefix, example: TR2_, TR3_ etc. |
| DI_ | Dimension refresh |
| FT_ | Fact table refresh |

Note: If many of these tasks are performed in the same package, it is worth to consider splitting the package in several packages.

1.3) Packagelogging

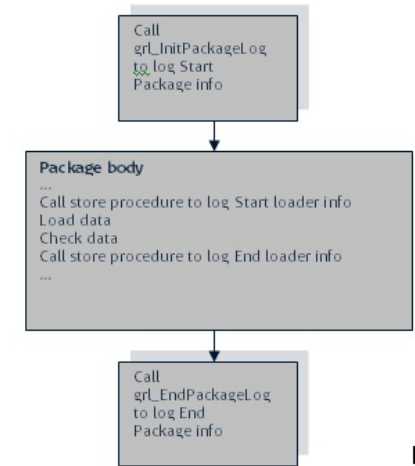
Every time a package is executed, the table etl.Grl_PackageLog is updated to log data about the executed package. For this purpose, two store procedures are called, the first at the at the beginning of the package execution: etl.grl_InitPackageLog, and the second at the end: etl.grl_EndPackageLog. The first one, logs information about of the package start, such as, the name and start time of the execution. The second one, logs information, such as, the end-time of the execution and a flag indicating whether the package was executed successfully or not.

More details on the packagelog can be found in "The MDW-database | The ETL-schema".

1.4) The structure of a package

A package is divided in different logical groups. The logical groups vary, depending on the schema and purpose of the package.

The general structure of a loader package is:



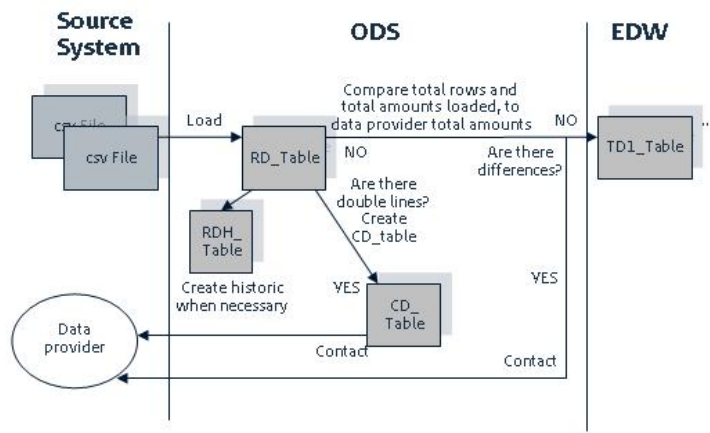
1.5) Naming tasks

The logic of the tasks in a package is criteria of the package developer, however, the titles in the tasks can use a prefix to describe very briefly the main purpose of the task. After the prefix, a very brief description of the task have to be written. The following table lists the prefixes that can be used in the IS tasks titles. The table is for inspiration only and if it makes more sense to omit the prefixes, then it's also allowed.

| Prefix or name | Task |
|--------------------------|-------------------------------------------|
| EXEC | Execute a store procedure |
| TRUN | Truncate a table |
| LOAD | Execute a load |
| UPDT | Update a table |
| SEND | Send an email |
| CHK | Check data |
| CALC | Calculate data |
| CONT | Count Rows in file or table |
| CONV | Convert data |
| SORT | Sort data |
| | |
| | |
| Source file name | Name of the file being loaded |
| Source/target Table name | Name of the source or target table in use |

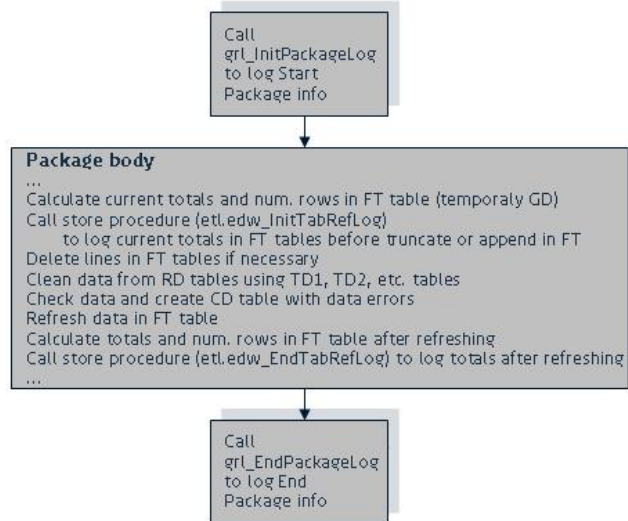
2) ODS-packages

When the data are copied to the RD tables, the first data check is made: double lines (because the data in the source files must be aggregated). For loaders controlling historic-data changes, such as LTD2, a RDH table is updated, registering these changes. If there are double lines, a table is updated with the prefix CD_, which is a prefix to identify tables for Checking Data purposes. The following diagram illustrates the types of tables (RD, RDH and CD) in the ODS schema, and its processes.



3) EDW-packages

The general structure of a data-transformation package is:



Revisioner

| Version | Godkendt | Revisions information |
|---------|------------|-----------------------|
| 1 | 11.02.2009 | |
| 1.1 | 08.03.2010 | |
| 1.2 | 17.08.2011 | |