CloneDB

Version 6.5.2

Table of Contents

1 Functionality

This package provides the 'CloneDB' feature. It clones a Znuny source database into a target database, possibly on a different RDBMS. This script clones a Znuny database into an empty target database, even on another database platform. It will dynamically get the list of tables in the source DB, and copy the data of each table to the target DB.

1.1 Usage

1.1.1 Cloning a Database

The goal of this example is to clone current database into another database engine.

Configure target database settings for this package (CloneDB).

The following parameters should be configured:

- TargetDatabaseHost
- TargetDatabase
- TargetDatabaseUser
- TargetDatabasePw
- TargetDatabaseType

Where *TargetDatabaseHost* is the host name or ip address of the target system, T*argetDatabase* is the name if the target database, *TargetDatabaseUser* is the name of the user that connects to the target database, *TargetDatabasePw* is the password of the user, be aware that the field is not masked and it is stored in plain text, *TargetDatabaseType* is the type of the target database (mysql, postgresql or oracle).

On *CloneDB::TargetDBSettings* for Oracle you might want to configure extra parameters like:

- TargetDatabaseSID
- TargetDatabasePort

If they are not set, the default values (XE and 1521 respectively) will be used.

And please be aware to include the needed settings on the *Config.pm* file:

```
$ENV{ORACLE_HOME}
$ENV{NLS_DATE_FORMAT}
$ENV{NLS_LANG}
```

- TargetDatabaseDriver

If not set, the default value "SQL Server" will be used.

If you don't have access to the graphical user interface of the source Znuny system, for example during a migration, you can also perform the configuration directly in *Kernel/Config.pm* like this:

```
Self->{'CloneDB::TargetDBSettings'} =  {
'TargetDatabaseHost' => 'localhost',
```

```
    'TargetDatabaseType' => 'mysql',
    'TargetDatabase'     => 'znuny_target',
    'TargetDatabaseUser' => 'znuny_target'
    'TargetDatabasePw'   => 'znuny_target',
    };
```

Configure source database settings for this package (CloneDB). The only one setting currently available is CloneDB::SkipTables that allows you to set a list of tables should be skipped on the clone DB process, this feature is useful for internal use tables like for example in Oracle with the 'htmldb_plan_table' that is generated when you click on the 'Explain' Tab on the Oracle Web interface. Please use just lower case for this setting.

Run the console command `bin/otrs.Console.pl Maint::Database::Clone`.

It strongly recommended to do a read and check test before actually migrate the data, this can be done with the `bin/otrs.Console.pl Maint::Database::Clone --dry-run`.

Note: After a dry run, you will need to empty the target database, as the script only works with an empty database.

Verify the result. This add-on provides a simple way to clone data into a new freshly created target database of choice. It can copy your data between supported databases:

- oracle
- mysql
- mariadb
- postgres

This requries a clean target database.

2 Requirements

2.1 Framework

The following OTRS framework versions are supported:

- 6.5.x

2.2 Add-ons

These add-ons are required:

- None ## Operating System

These operating systems are required for this add-on:

- None

2.3 Third Party Software

These additional software is required for this package:

- None

3 Installation

The following instructions show how to install the package. There are two possibilities. Either via the admin interface or by using a shell command.

3.1 Admin interface

With the URL http://localhost/otrs/index.pl?Action=AdminPackageManager you access the package manager where the package can be installed and updated. The user who like to perform this needs to be member of the OTRS admin group.

3.2 Shell

As an alternative you can use the OTRS command line package manager to install the package. Please execute this as the OTRS user.

```
shell> bin/otrs.PackageManager.pl -a install -p /path/to/CloneDB-6.5.2.opm
```

3.3 Remarks

If unexpected problems occur after the installation it is recommended to restart the web server. Using mod_perl sometimes leads to generic loading problems and does not belongs to OTRS in general.

4 Configuration

4.1 System Configuration Options

CloneDB::BlobColumns

This setting specifies which table columns contain blob data as these need special treatment.

CloneDB::CheckEncodingColumns

Specifies which columns should be checked for valid UTF-8 source data.

CloneDB::LogFile

Log file for replacement of malformed UTF-8 data values.

CloneDB::SkipTables

List of tables should be skipped, perhaps internal DB tables. Please use lowercase.

CloneDB::TargetDBSettings

4.2 Configure target database settings for this package (CloneDB).

- TargetDatabaseHost
- TargetDatabase
- TargetDatabaseUser
- TargetDatabasePw
- TargetDatabaseType

Where "TargetDatabaseHost" is the host name or IP address of the target system, "TargetDatabase" is the name of the target database, "TargetDatabaseUser" is the name of the user for the target database, "TargetDatabasePw" is the password of the user, be aware that the field is not masked, and it is stored in plain text, "TargetDatabaseType" is the type of the target database (MySQL, Postgres or oracle).

```
Notes on Oracle:

On 'CloneDB::TargetDBSettings' for Oracle, you might want to configure extra parameters like

 - TargetDatabaseSID
 - TargetDatabasePort

If they are not set, the default values (XE and 1521 respectively) will be used.
And please be aware to include the needed settings in the Config.pm file:

 - $ENV{ORACLE_HOME}
 - $ENV{NLS_DATE_FORMAT}
 - $ENV{NLS_LANG}
```

- TargetDatabaseDriver

If you don't have access to the graphical user interface of the source Znuny system, for example during a migration, you can also perform the configuration directly in `Kernel/Config.pm` like this:

```
$Self->{'CloneDB::TargetDBSettings'} = {
  'TargetDatabaseHost' => 'localhost',
  'TargetDatabaseType' => 'mysql',
  'TargetDatabase'  => 'znuny_target',
  'TargetDatabaseUser' => 'znuny_target'
  'TargetDatabasePw'  => 'znuny_target',
};
```

Configure source database settings for this package (CloneDB). The only one setting currently available is CloneDB::SkipTables that allows you to set a list of tables to be skipped during the cloning. This feature is useful for internal use tables, like for example in Oracle with the 'htmldb_plan_table' that is generated when you click on the 'Explain' Tab on the Oracle Web interface. Please use just lower case for this setting.

It strongly recommended doing a read and check test before actually migrating the data, this can be done with the "-dry-run" option as: `su - $OTRS_USER -c bin/otrs.Console.pl Maint::Database::Clone --dry-run`. Remember to drop all tables from the target database, after a dry run or failure.

4.3 Other System Configuration Options

4.3.1 CloneDB::BlobColumns

This setting specifies which table columns contain blob data as these need special treatment.

4.3.2 CloneDB::CheckEncodingColmns.

Specifies which columns should be checked for valid UTF-8 source data.

4.3.3 CloneDB::LogFile

Log file for replacement of malformed UTF-8 data values.

4.3.4 CloneDB::SkipTables

List of tables should be skipped, perhaps internal DB tables. Please use lowercase.

4.3.5 CloneDB::TargetDBSettings

Settings for connecting with the target database.

5 Changelog

6.5.2 / 2023-10-02 15:21:58 +0200 Manual updated

6.5.1 / 2023-02-20 14:33:22 +0100 Initial release for Znuny 6.5.

6.4.1 / 2022-07-29 09:19:33 +0200 Initial release for Znuny 6.4.

6.3.1 / 2022-03-31 19:19:34 +0200 Ported to 6.3

6.2.1 / 2021-11-09 11:00:24 +0100 Ported to 6.2

6 Manifest

This is the list of all files from the package with the file system permissions.

- (660) Kernel/Config/Files/XML/CloneDB.xml
- (660) Kernel/Language/bg_CloneDB.pm
- (660) Kernel/Language/de_CloneDB.pm
- (660) Kernel/Language/es_MX_CloneDB.pm
- (660) Kernel/Language/es_CloneDB.pm
- (660) Kernel/Language/hu_CloneDB.pm
- (660) Kernel/Language/it_CloneDB.pm
- (660) Kernel/Language/ja_CloneDB.pm
- (660) Kernel/Language/nb_NO_CloneDB.pm
- (660) Kernel/Language/pl_CloneDB.pm
- (660) Kernel/Language/pt_BR_CloneDB.pm
- (660) Kernel/Language/pt_CloneDB.pm
- (660) Kernel/Language/ru_CloneDB.pm
- (660) Kernel/Language/sr_Cyrl_CloneDB.pm
- (660) Kernel/Language/sr_Latn_CloneDB.pm
- (660) Kernel/Language/zh_CN_CloneDB.pm
- (660) Kernel/Language/zh_TW_CloneDB.pm
- (660) Kernel/System/CloneDB/Backend.pm
- (660) Kernel/System/CloneDB/Driver/Base.pm
- (660) Kernel/System/CloneDB/Driver/mysql.pm
- (660) Kernel/System/CloneDB/Driver/postgresql.pm
- (660) Kernel/System/CloneDB/Driver/oracle.pm
- (660) Kernel/System/Console/Command/Maint/Database/Clone.pm
- (660) scripts/test/Console/Command/Maint/Database/Clone.t

7 Contact

7.1 EMEA

Znuny GmbH
Marienstraße 18
10117 Berlin
Germany
P +49 (0) 30 60 98 54 18-0
F +49 (0) 30 60 98 54 18-8
E info@znuny.com

7.2 AMER

Znuny Inc.
171 C Avenue, Suite C
Coronado, 92118 CA
United States
P: +1 949 431 2599
F: +1 949 431 2477
E info@znuny.com

7.3 Switzerland

Znuny Swiss GmbH
Martinsbruggstrasse 35
9016 St. Gallen
Switzerland
P +41 (0) 71 588 03 39
F +41 (0) 71 588 01 86
E info@znuny.ch