

Arcsynchro - Archive Synchronization Tool

The application **arcsynchro** is an independent program (the file *arcsynchro.exe* is placed in the subdirectory **Bin** of the [installation directory](#)). It must be started manually from the command line. It requires information on source and target database for starting, and then it synchronizes the databases (target database is updated by the records from source database). The [D2000 Server](#) process or other processes need not to be running except for the database servers.

Contents

- [Declaration](#)
- [Parameters](#)
- [Parameters for depository databases](#)
- [Other parameters](#)
- [Description](#)
- [Example](#)
- [Return codes](#)
- [Arcsynchro run directly by archive](#)

Declaration

```
arcsynchro [optional parameters] source target time_start time_end [id|+id|-id|mask]

arcsynchro [optional parameters] [/TD time_delta] /A source target [id|+id|-id|mask]

arcsynchro [optional parameters] [/TD time_delta] /AN source target

arcsynchro [optional parameters] [/TD time_delta] /AX source target [id|+id|-id|mask]

arcsynchro [optional parameters] /A+ source target [id|+id|mask]

arcsynchro /CLD source target
```

Parameters

/S U sou rce _us er	User name for <i>source</i> (required if it is other than standard).
/SP sou rce _p ass word	User password for <i>source</i> (required if it is other than standard).
/TU tar get _us er	User name for <i>target</i> (required if it is other than standard).
/TP tar get _p ass word	User password for <i>target</i> (required if it is other than standard).

source	TNS (if <i>source</i> is an Oracle database) or DSN (if <i>source</i> is a database that is available through ODBC configured) of an archive containing valid archive records.										
target	TNS or DSN of archive with missing records.										
time_start	Beginning of the time interval for completing missing archive records.										
time_end	End of the time interval for completing missing archive records.										
id	Historical value identifier (number). If id=113, then corresponding archive table is DT0000113 (names of archive tables which are being processed are being listed when the program runs).										
+id -id +id -id	<p>Historical value identifier (number), from which (+id), upto which (-id) or for which interval (+id -id) the synchronization will be performed. If e.g. +113 is specified, then historical values with HOBJ=113 and higher will be synchronized (tables DT0000113, DT0000114 etc).</p> <p>An interval can be specified using +id -id, e.g. +1 -999 will synchronize the historical values with HOBJ between 1 and 999.</p> <p>Note: This parameter can be used to parallelize the synchronization of archives (e.g. when loading data to a new archive). Several arcsynchro processes can be run in parallel. The first arcsynchro uses parameters +1 -999, the second +1000 -1999 etc. To separate the log files of multiple arcsynchro processes, we recommend using also the parameter LOGDT.</p> <p>Furthermore, this parameter can be used after the interruption of arcsynchro work (e.g. due to loss of connectivity to the database server) and resume from a specific archive object (since the archive objects are synchronized in the order of their numeric identifiers).</p>										
mask	<p>Mask for the archive variable name. The mask can contain special characters (? or *) which allow to synchronize selected variables. Mask is not case sensitive.</p> <p>Example: The variables H11_Edo, H11re, h111Element matches the mask H11?E*. The variables H11abEdo, H11Edo, H12_edo doesn't match the mask.</p>										
/A	<p>Automatic archive synchronisation based on the information on times when archives were broken down. The times are stored in the destination archive (the table ARC_HOLE, since the version D2000 v7.00.031).</p> <p>When starting, the archive generates new log into the table ARC_HOLE and periodically writes into it. The structure of the table ARC_HOLE is as follows:</p> <table border="1"> <thead> <tr> <th>Column</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Id</td><td>Row identifier.</td></tr> <tr> <td>From_time</td><td>Beginning of archive hole (blank space).</td></tr> <tr> <td>To_time</td><td>End of archive hole.</td></tr> <tr> <td>Status</td><td> <p>Hole status:</p> <ul style="list-style-type: none"> 0 - current hole (<i>From_time</i> is still increasing) 10 - holes in archive as well as depository database (if creating depository databases is enabled) that need to be synchronised 15 - hole in archive needs to be synchronized, hole closed in depository database (non-standard status) 20 - hole closed in archive, depository database has not been synchronized 25 - holes closed in archive as well depository database 30 - hole, that has been closed, but no values were inserted at synchronization -1 - blocked hole (will not be synchronised) </td></tr> </tbody> </table> <p>If both the archive and depository database are synchronized, the synchronisation is to be performed for the holes with Status=10. If the archive is only synchronized, the synchronisation is to be performed for the holes with Status=15. If depository database is only synchronized, the synchronisation is to be performed for the holes with Status=20.</p> <p>After the synchronisation has been finished successfully (if there is no hole in the source archive that contains the hole of the destination archive and if there is inserted at least one record during synchronisation), there will be set:</p> <ul style="list-style-type: none"> Status=15 - there was synchronized depository database only Status=20 - there was synchronized the archive only Status=25 - both the archive and depository database were synchronized <p>If no record was inserted, there will be set Status=30 - then the arcsynchro will not attempt to synchronise the same hole again.</p>	Column	Description	Id	Row identifier.	From_time	Beginning of archive hole (blank space).	To_time	End of archive hole.	Status	<p>Hole status:</p> <ul style="list-style-type: none"> 0 - current hole (<i>From_time</i> is still increasing) 10 - holes in archive as well as depository database (if creating depository databases is enabled) that need to be synchronised 15 - hole in archive needs to be synchronized, hole closed in depository database (non-standard status) 20 - hole closed in archive, depository database has not been synchronized 25 - holes closed in archive as well depository database 30 - hole, that has been closed, but no values were inserted at synchronization -1 - blocked hole (will not be synchronised)
Column	Description										
Id	Row identifier.										
From_time	Beginning of archive hole (blank space).										
To_time	End of archive hole.										
Status	<p>Hole status:</p> <ul style="list-style-type: none"> 0 - current hole (<i>From_time</i> is still increasing) 10 - holes in archive as well as depository database (if creating depository databases is enabled) that need to be synchronised 15 - hole in archive needs to be synchronized, hole closed in depository database (non-standard status) 20 - hole closed in archive, depository database has not been synchronized 25 - holes closed in archive as well depository database 30 - hole, that has been closed, but no values were inserted at synchronization -1 - blocked hole (will not be synchronised) 										
/AN	The same functionality as /A , but actual patching is not performed - arcsynchro exits after displaying the list of holes that need to be synchronized. This parameter enables to find out the number and sizes of holes that need to be synchronized.										

/AX	<p>The same functionality as /A but value of Status in ARC_HOLE table is not changed after the hole had been patched.</p> <p>Note: This parameter can be used for patching redundant archives with 3 or more instances. When patching the hole in database1 using database2 and database3, two arcsynchro would be activated:</p> <pre>arcsynchro /AX database2 database1 arcsynchro /NU /A database3 database1</pre> <p>First line patches the hole in database1 with data from database2 but keeps the hole marked as desynchronized. Second line patches (only inserts, does not update) the hole in database1 with data from database3 and marks the hole as patched. Usually only one arcsynchro command is needed. Two arcsynchro can handle the situation when also database2 and database3 contain incomplete data (i.e. archives [2] and [3] were down for some period during the longer time interval while archive [1] was down).</p>
/A+	<p>For every Historical value the whole history depth will be synchronised (as configured on the object) up to the present time. This parameter can be used for duplicating the whole archive database into a new database. Normally this parameter can be replaced by suitably chosen <i>time_start</i> and <i>time_end</i> parameters (<i>time_start</i> must be sufficiently far in the past to cover maximum configured history depth), but if the source archive is configured not to delete old values, this would mean that also old values would be synchronized.</p> <p>Note: Configuration of history depth is queried from source or target database (see the description of /TAD parameter).</p>
/ATE /ATD /ATA	<p>Archived objects with enabled archiving (/ATE, default), disabled archiving (/ATD) or all (/ATA) will be synchronized. This parameter can be used for duplicating the whole archive database into a new database if copying of old archive objects with disabled archiving is desired. For these objects, the corresponding tables will be created in the target database if they didn't exist.</p>
/TD time_delta	<p>The size of the hole surroundings in the archive that is to be synchronised.</p> <p>When writing into the archive, communication can send values with the older or latter timestamp than is the current time. After the failure of the archiving process, there are missing latter and older data surrounding the hole. Therefore, it is possible to define the size of the hole surrounding using the parameter <i>time_delta</i> (in seconds), that is to be synchronised. Default value <i>time_delta</i> = 10.</p>
/FD future_delta	<p>The size of the time interval from the end of the hole in the archive, which will also be synchronized.</p> <p>When writing to the archive, data with a future time (e.g. production planning) can also be inserted (usually from a script). After the failure of the archiving process, newer data such as the time of the end of the hole is also missing. Therefore, by specifying the <i>future_delta</i> parameter, it is possible to set the time of data added to the end of the hole (in seconds), which will also be synchronized. Default <i>future_delta</i> = 0.</p>
/NU	<p>If the target archive database already contains the value with timestamp being inserted (therefore insert fails) the value will not be updated. This parameter is meant to speed-up the synchronisation if the target archive database contains only holes and not data with different values than the source database.</p>
/U	<p>Updates the tables structures in the archive database. If at least one of the tables in the archive database is not actual (e.g. ARCHIV_DEF, ARC_HOLE, UTC_OFFSET tables), the Arcsynchro utility runs without the parameter /U is to be terminated with an error message. The Arcsynchro utility runs with the parameter updates the structures of the tables in the archive database (it may cause some problems if the new Arcsynchro utility is used for patching holes in an older application that could not work with the archive database upgraded by using this parameter).</p>
/UX	<p>Work with old (unupgraded) structures of tables in the archive database. This parameter enables to synchronize data from older D2000 applications (e.g. pre-UTF8 D2000 version 8.0) without need to upgrade the old archive database. This is convenient e.g. when preparing a migration to a new D2000 version using a different database platform for D2000 Archiv.</p>
/UF	<p>Arcsynchro first attempts to update the value. If no row is modified as a result of update, then insert is performed. Parameter is implemented only for ODBC version.</p> <p>Note: parameter can be used for PostgreSQL archives to minimize transaction ID (XID) generation when used against a DSN with "Level of rollback on errors" set to Transaction.</p> <p>Note: for PostgreSQL starting with version 9.5 it is possible to use parameter /UP to minimize XID generation.</p> <p>Note: identical functionality in archive is activated by archive parameter SelectBeforeUI.</p>
/UP	<p>Parameter activates usage of SQL command "UPsert" (combination of Insert and Update SQL commands), which is supported by PostgreSQL database starting with version 9.5. In this case insertion of values into archive and depository databases will be performed by command INSERT .. ON CONFLICT .. DO UPDATE. Parameter is implemented only for ODBC version.</p> <p>Note: parameter can be used for PostgreSQL archives to minimize transaction ID (XID) generation when used against a DSN with "Level of rollback on errors" set to Transaction.</p> <p>Note: for versions of PostgreSQL older than 9.5 it is possible to use parameter /UF to minimize XID generation.</p> <p>Note: identical functionality in archive is activated by archive parameter Upsert.</p>
/PO	<p>Only primary archives will be synchronized (not the statistical and evaluated ones).</p>
/TID tid	<p>The parameter is only allowed if the Arcsynchro - Archive Synchronization Tool#id parameter (data synchronization of one object with a particular identifier) is specified.</p> <p>In that case, values of the object with the Arcsynchro - Archive Synchronization Tool#id identifier read from the source database are inserted into the target archive database into the tables of the object with identifier <i>tid</i>. If the data in the depository databases is synchronized, the identifier of the <i>id</i> object will be replaced by the <i>tid</i> identifier when data is inserted into the target depository databases.</p> <p>Note: The /TID parameter is useful when merging applications and their archives if there is a change in HOBj of objects (the object ID in the source application and the source archive is different from the target application/archive).</p>
/LOGDT	<p>Instead of using the default log file name <i>arcsynchro.log</i> arcsynchro will create a log file with a timestamp in its name <i>arcsynchro yyyy-mm-dd hh-mm-ss.log</i>. For a reason to use this parameter please see this note.</p>

Parameters for depository databases

/ST trez or_ dsn	<p>DSN of source Sybase depository database.</p> <p>The parameter can be used just once. By default, it also enables the parameter <i>/TAD</i> (i.e. only values of objects, the definitions of which are in target archive database, are to be read from the depository database).</p> <p>Note: If the parameter <i>/ST</i> is used, the parameter <i>source</i> (source archive DSN) will be ignored.</p>
/ST PG trez or_ name	<p>Name of source PostgreSQL depository database. Usually in <i>appname_TREZOR_n</i> format, e.g. Test_TREZOR_11.</p> <p>The parameter can be used just once. By default, it also enables the parameter <i>/TAD</i> (i.e. only values of objects, the definitions of which are in target archive database, are to be read from the depository database).</p> <p>Note: If the parameter <i>/STPG</i> is used, the parameter <i>source</i> (source archive DSN) will be used to connect to the PostgreSQL depository database (in order not to create a separate DSN for each PostgreSQL depository). Therefore <i>source</i> must point to the same PostgreSQL server where the depository database is located.</p>
/TT trez or_ dsn	<p>DSN of destination depository database.</p> <p>The parameter may be repeated several times for destination depository database on Sybase (e.g. for synchronisation of the current and previous depository database: <i>/TT app.curr_trezor /TT app.prev_trezor</i>).</p> <p>Note: If the first database cannot be opened, the arcsynchro will be stopped with an error. Error in opening of the second or another database is to be ignored (to allow standard use with current and previous depository databases, while the previous one cannot be available at all).</p> <p>For target depository database on Oracle, the parameter may be used in case that depository databases and the archive database are not stored in the same place (use the parameter <i>/TTO</i> otherwise). The parameter can be used only once. The user name used for connecting the database is specified by the parameter <i>target_user</i>, the user name may be defined by the parameter <i>/TTP trezor_password</i>.</p>
/ST O trez or_id	<p>For source depository database on Oracle: data from depository database <i>trezor_id</i> will be synchronised. Parameter can be used for Oracle depository databases stored along with the archive in the same database. Parameter enables filling of target database, or depository databases (Oracle, PostgreSQL, Sybase) from Oracle depository database.</p> <p>Examples:</p> <ul style="list-style-type: none"> • <i>/STO 16</i> - synchronisation from depository database 16 • <i>/STO 16_S01</i> - synchronisation from depository database 16, depository database segment No.1 <p>Note: Data can be read only from mounted depository databases (current, previous and older depository databases mounted by the Tell command MOUNT_TREZOR).</p>
/TTO	<p>For destination depository database on Oracle: the parameter enables synchronization of depository databases. This parameter can be used for depository databases stored along with the archive in the same database. If the depository databases are stored in different database, there must be used the parameter <i>/TT trezor_dsn</i>.</p> <p>Note: Data can be written only to depository databases mounted for writing (current, previous and older depository databases mounted by the Tell command MOUNT_TREZOR WRITE).</p>
/PTO	<p>Synchronization of depository databases (exclusive of the archive) in the destination database. The parameter <i>target</i> will be ignored if the parameter <i>/TT trezor_dsn</i> is used and the parameter <i>/TAD</i> is not used.</p>
/TT P trez or_ pas sw rd	<p>Password for connection to the destination depository database.</p>
/TAD	<p>Arcsynchro uses the definitions of historical objects from the destination database. Although the definitions of historical values from the destination database are used by default, the parameter is required in case that the number of depository database segments in the destination Oracle archive is different from the number in the destination archive and data for depository databases are synchronized as well. Without the parameter, the archive either stores data in the depository database segment 02 (if the number of segments in the source archive is less than the number of segments in destination archive) or cannot find any segment to store the data in (if the number of segments in the source archive is greater than the number of segments in destination archive).</p>
/TT PG 0 mas k0 /TT PG N mas kN	<p>For destination depository database on PostgreSQL: the parameter enables synchronization of depository databases.</p> <p>If only a single depository database segment is configured (<i>TrezorCountSegments=0</i>), only the parameter <i>/TTPG0 mask0</i> has to be specified. Parameter <i>mask0</i> has to be equal to the value of parameter <i>PG_TrezorName0</i>.</p> <p>If several depository database segments are configured (<i>TrezorCountSegments>0</i>), also the parameter <i>/TTPGN maskN</i> has to be specified. Parameter <i>maskN</i> has to be equal to the value of parameter <i>PG_TrezorName</i>.</p> <p>Note: Data can be written only to depository databases mounted for writing (current, previous and older depository databases mounted by the Tell command MOUNT_TREZOR WRITE).</p>

/FM path	<p>For PostgreSQL depository database on PostgreSQL from version 9.5: data is not entered into the depository databases via ODBC, but a file is created in the <i>path</i> directory where the data will be in the text form. After inserting the number of rows that corresponds to the startup parameter <i>/CM</i> (default 1000 but recommended to increase to 1000000 if <i>/FM</i> is used), a foreign table mapped to this file is created via PostgreSQL extensions 'file_fdw', and the data is merged into the depository database (upsert).</p> <p>This procedure bypasses slowness of ODBC, which processes rows one at a time. It is necessary for the <i>path</i> directory to be readable for the PostgreSQL database (i.e., arcsynchro must run on the same computer as the database).</p> <p>By using the <i>/ FM</i> parameter, data transfer has been accelerated approximately 3 times (from 2400 to 7050 rows per second) in a specific case.</p>
-------------	---

Other parameters

/C LD	<p>Copies the contents of the table LOG_DATA.</p> <p>The copying is necessary when adding a new instance of D2000 Archiv to the existing application or during migration to a new database (e.g. when migrating from lower to higher version of Sybase, or from Sybase to Oracle platform and vice-versa). Table LOG_DATA contains information about archive's transition to monotonic time and about the start of on-change archiving. If older data are to be copied to archive database (from older archive database), it is necessary to copy also the contents of LOG_DATA table and only afterwards to start archiving. Otherwise old data may be misinterpreted and possibly damaged during recalculations.</p> <p>Copying of the contents of the table LOG_DATA must be performed before D2000 Archiv starts writing data to the new database.</p> <p>Recommended procedure is following:</p> <ol style="list-style-type: none"> 1. Create the new archive database (Sybase or MSSQL platforms) or archive tablespace (Oracle platform). 2. Run D2000 Archiv with parameter <i>/RO</i> (read only). The tables for archived data will be created, but no values will be inserted. 3. Stop D2000 Archiv as soon as tables are created. 4. Run: <pre>arcsynchro /CLD source target,</pre> <i>source</i> is DSN of the old archive database with historical data, <i>target</i> is DSN of the new archive database with empty tables. 5. At this point it is possible to run D2000 Archiv (start of normal archiving to a new archive database). 6. Copy the historical data from the old database into the new database <pre>arcsynchro source target time_start time_end,</pre> <p>Note: the row 'STOP' saving the time of the last stop of D2000 Archiv will not be copied to target database.</p>
/D DD	<p>Detailed debug information.</p> <p>After the parameter is enabled, detailed debug information will be shown on the screen as well as written into the <i>arcsynchro.log</i> log file.</p>
/D C c o u n t	<p>Setting the parameter to non-zero value <i>count</i> will cause letters 'I' or 'U' shown on the screen as well as written into the <i>arcsynchro.log</i> log file after inserting or updating <i>count</i> values.</p> <p>Note: Letters are displayed continuously on the screen but only after processing of one object in log file.</p>
/C M c o u n t	<p>Commits in archive database will be activated after <i>count</i> inserted/updated values. Default value of this parameter is 1000.</p>
/X A D C c o n d i t i o n	<p>Supplementary condition for the ArchivDef table (a table with definitions of archive objects in the archive).</p> <p>Example of use: when using arcsynchro to copy data from depository segments (e.g. from an Oracle database) to depository segments (e.g. PostgreSQL database) it is possible to specify <i>/XADC "TRZ_SEG=2"</i>. Only objects that are in depository segment no. 2 will be synchronized. This can speed up data copying (the source depository segment is not queried for object data that is in other depository segments).</p>

Description

The **Arcsynchro** utility is used for synchronization of archive databases when one of the archives (*target*) is dropped-out. The archive will connect the archive database (*source*) containing required data within the defined time interval *<time_start, time_end>* and the date will be copied to the target archive.

Archived values, which are already stored in the target archive will be overwritten (UPDATE).

Since the D2000 system, version 7.00.031, there has been supported the **automatic synchronisation** mode, that gains required time_start/time_end directly from the archive (see the parameter /A mentioned above).

For version 7.01.025 and newer: the format of beginning and end times (*time_start, time_end*) is 'YYYY-MM-DD HH:MI:SS' and it is a local time (time of the user).

For version 7.01.024 (Release 65) and older: the format of beginning and end times (*time_start, time_end*) depends on particular database and the operating system (see the example below) and entered time is directly compared to time in archive database.

For an Oracle database, the time format is to be set to YYYY-MM-DD HH24:MI:SS.

After starting, **Arcsynchro** displays synchronization info on the monitor as well as writes it into the file **arcsynchro.log** in the current directory. The file contains extra information on the time when the program is started and closed as well as time stamps of all logs.

For Sybase platform: There must be specified DSN for all depository databases that is to be synchronized.

For Oracle platforms: DSN (arcsynchro.exe) or TNS (arcsynchro_ora.exe) specifies the target database.

Example

Sybase database under Windows operating system:

```
arcsynchro.exe dbsrvmain dbsrvbackup "2001-03-30 15:00:00" "2001-03-30 15:30:00"
```

Oracle database:

```
arcsynchro_ora.exe /SU myapp_archiv /TU myapp_archiv dbsrvmain dbsrvbackup "2004-03-13 10:40" "2004-03-14 03:12:25"
```

Oracle database with synchronisation of depository databases:

```
arcsynchro_ora.exe /TTO /TAD /SU myapp_archiv /TU myapp_archiv /A dbsrvA dbsrvB
```

Sybase database, two depository databases (current and previous ones) are to be synchronized:

```
arcsynchro.exe /TT TrzCurr /TT TrzPrev dbsrvmain dbsrvbackup "2004-03-13 10:40" "2004-03-14 03:12:25"
```

Oracle database, all available depository database (current, previous and older depository databases mounted for writing) are to be synchronized:

```
arcsynchro_ora.exe /TTO /SU myapp_archiv /TU myapp_archiv dbsrvmain dbsrvbackup "2004-03-13 10:40" "2004-03-14 03:12:25"
```

Copying data from old Sybase depository database *TrzSrc* to Oracle archive *TrzDstArc* and to Oracle depository database in this archive as well:

```
arcsynchro /TTO /ST TrzSrc unusedDSN TrzDstArc "2006-07-27 07:00:00" "2006-07-27 08:00:00"
```

Copying data from old Sybase depository database *TrzSrc* to Oracle archive *TrzDstArc* and to Oracle depository database in this archive as well, supposing depository database is configured to store data in several depository database segments (parameter /TAD is necessary):

```
arcsynchro /TAD /TTO /ST TrzSrc unusedDSN TrzDstArc "2006-07-27 07:00:00" "2006-07-27 08:00:00"
```

Automatic patching of archive database and depository databases on PostgreSQL, depository database segments are not used and only primary historical objects are patched:

```
arcsynchro /LOGDT /PO /TPG0 MyAPP_TREZOR_#ID# /A MyApp.Archiv.Arc2 MyApp.Archiv.PG
```

Return codes

0	Successful completion of synchronisation.
1	Error in program parameters.
2	Error connecting to source database specified by parameter Arcsynchro - Archive Synchronization Tool#source .
3	Error connecting to source depository database specified by parameter /ST.
4	Error connecting to target database specified by parameter Arcsynchro - Archive Synchronization Tool#target .
5	Error connecting to separate target depository database on Oracle platform specified by parameter /TT trezor_dsn.
6	Error connecting to first target depository database on Sybase platform specified by first parameter /TT trezor_dsn.
7	The conversion to time slices in Arcsynchro - Archive Synchronization Tool#source database was started but not finished.
8	The conversion to time slices in Arcsynchro - Archive Synchronization Tool#target database was started but not finished.

9	Target database is not on Oracle platform and parameter /TTO was used.
10	Error reading list of depositories from Arcsynchro - Archive Synchronization Tool#target database on Oracle platform using the parameter /TTO.
11	Error reading list of depositories from separate depository database /TT trezor_dsn on Oracle platform using also the parameter /TTO.
12	Error reading parameters of source depository database specified by parameter /ST.
13	Error during data synchronisation (e.g. broken connection to database).

Note 1

If **arcsynchro.exe** (ODBC version) is used for accessing the Oracle database Oracle9i, the ODBC version MUST be higher than 9.2.0.0 (part of Oracle ODBC installation). The version 9.2.0.0 generates the errors when working with timestamps and returns zero number of row for specified time period, although the database contains any rows for the period.

We recommend to use the patch for ODBC drivers 9.2.0.6 and higher.

Warning: Oracle 9.2.0.6 patchset (approx. 200MB) does not contain ODBC 9.2.0.6 patch (approx. 2MB) - it must be downloaded from the Oracle website.

Note 2

If you use the /A parameter (automatic synchronisation), the archive hole in target database will not be marked as patched if parameter id or mask was specified. If parameter +id was specified or no mask was specified and synchronisation is successful, archive hole will be marked as patched (status=20).

Note 3

If error "ORA-01555 Snapshot too old" appears during synchronisation of archives based on Oracle database, it is necessary to change the Undo Retention parameter of database and repeat synchronisation. This error can appear especially during synchronisation of archives which contain large structured variables.

Arcsynchro run directly by archive

Starting with version 9.00.021, a new feature has been implemented in the process **D2000 Archiv**. It can run arcsynchro immediately after it has been started and initialised. A new archive parameter **ArcsynchroAuto** is used to enable this feature. It must contain full path and parameters for the arcsynchro utility or for a command file (.bat, .cmd), which runs arcsynchro.

The parameter **ArcsynchroAuto** is used only if the archive is running as an **instance (shadow) process**. (If it is not the shadow, we presume it is the only archive, therefore there is no other archive to synchronise with it. The configuration of the system with redundant kernels, each of them having a locally run archive, is not supported, because it is considered improper.).

Examples:

C:\D2000\D2000.exe\bin\arcsynchro.exe /PO /A MyApp.Arc2 MyApp.Archiv

C:\D2000\D2000.exe\bin\arcsynchro_ora.exe /PO /TTO /TAD /SU myapp_archiv /TU myapp_archiv /A Arc2 Arc1 (on Oracle platform, also depository databases are synchronised)

In both cases the parameter /PO is used to synchronise only primary archives, because all statistical and evaluated archives will be recalculated later. Even before running arcsynchro, the archive reads a list of unpatched archive holes from the database (rows from the table **ARC_HOLE**, where Status=10). After running arcsynchro the archive periodically checks whether the status of these rows has been changed.

If the **Status** of the archive hole changes to value -1 (blocked hole) or 30 (no values were inserted at synchronization), the archive only removes the archive hole from the list of monitored holes.

If the **Status** of the archive hole changes to value 20 (closed hole in archive) or 25 (closed in archive as well depository database), the archive performs these activities:

- It queries the time of start and end of hole (using the table **ARC_HOLE**) and enhances the interval by **ArcsynchroTimeDelta** seconds (because of existence of delayed values) to < **From_time - ArcsynchroTimeDelta** ; **To_time + ArcsynchroTimeDelta** >.
- For all primary archived objects it queries the archive database whether any values have been inserted to archive hole. If it is so, it clears the cache of a primary archived object so that it does not contain information older than the end of patched hole.
- For all statistical and evaluated archived objects the cache is also cleared and a recalc for enhanced time interval is started. If the archive parameter **ArcsynchroBackground** is set to 1, the recalc is performed by an auxiliary calc task, otherwise the recalc is performed by main archive write task(s).

This functionality enables the archive to "postinitialize" after the archive hole has been patched and - if new values of primary archive objects have been inserted - read them and take them into account (otherwise it would only rely on values read during the startup).



Blog

You can read blogs about Arcsynchro utility:

- [Arcsynchro and PostgreSQL depositories](#) (Slovak only)
- [Migration of depositories to PostgreSQL - practical experience](#) (Slovak only)