

Replication Services

Installation of Replication Option	21.2
Configuration and Administration	21.2
Understanding the Replication Process	21.8
ERP2020 4D Slave Server	21.21
SOAP-Connectivity between Master and Slave	21.28

The ERP2020 system has several Replication Options. These options allow an ERP2020 Server to periodically export data to a remote slave server via an intermediate FTP host. The slave database server may be another 4D Server, an Oracle Database Server, or a MYSQL Server. During the replication process the ERP2020 installation acting as the MASTER exports data to one or more SLAVE

databases which in-turn import and integrate the data. This process is automatic and unattended. This chapter explains the operation of replication.

Installation of Replication Option

The replication option is installed by the designer as part of implementation. During the installation the designer also creates replication-control records in the MISCDATA TABLE. These records define the parameters related to the export of data from the Master and the import into the Slave. Each replication option (4D, MYSQL, ORACLE) has a separate MISCDATA record. Since these records are stored in the data-file, the replication option is therefore enabled via the **data-file** and **NOT by structure file**.

Configuration and Administration

The Replication-Control Records (in the MISCDATA TABLE) created by the designer have an ID of - 999 and the field NAME is set to REPLICATION4D, REPLICATIONMYSQL, and/or REPLICATIONORACLE, corresponding to the 4D to 4D, 4D to MYSQL and 4D to Oracle replications respectively. Once these records are created (by the Designer), the Administrator defines other configuration parameters such as the intermediate FTP HOST, FTP User Name and Password, the path to the FreeZip programs on the server-side, and the local directories where the server will create the data before exporting it to the intermediary FTP Server. The ZIP programs (zip.exe and unzip.exe) are used to compress the data on the server-side before transferring it to the FTP host. Similarly the UNZIP.exe file is used on the Slave side to un-compress the ZIP file. Replication data is in the 4D, MYSQL, or ORACLE format depending upon the nature of the export requested by the Replication Record(s). All or any combination of replications options may be selected. The limiting factor being the burden put on the 4D Server responsible for the replication.

The MISCDATA record for the replication option is shown in figure 21.1 on page 21.3.

STR_ID	-999	00/00/00
Name	ReplicationData4D	
Customer		
Replication Folder	C:\ERPRep	
Freezip directory	C:\FREEZIP	
Control Parameters		
Delay Ticks	0	
FTP Host	erp2020.com	Get Printer List
User Name	stsrep	
Password	stsrep	
Email Recipient	ramesh@erp2020.com, saeed@erp2020.com, ahsan@erp2020.com	
Export-File name delimiter { }		
Accelerated Duty Cycle	0	
	<input type="checkbox"/> Void	Error check

FIGURE 21.1 4D replication configuration

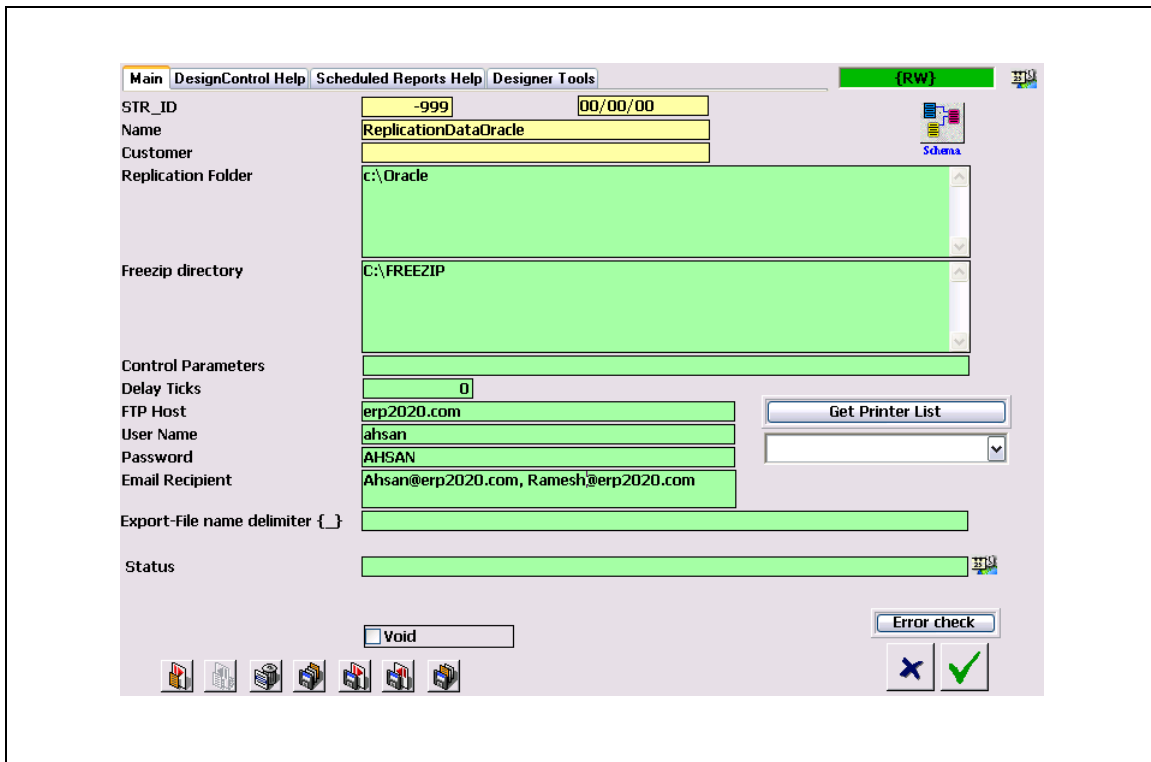


FIGURE 21.2 Oracle Replication configuration

Replication-Record form-fields:

- Replication Folder** This field is used to define the local-path of the folder to which the server will first export the data-files (later to be sent to the intermediary FTP server).
- Freezip Directory** This field is used to define the absolute-path to the FreeZIP.exe and UNZIP.exe files which will be used to compress the contents of the local directory defined in WORK1 and then, in the case of a 4D Slave, un-compresses the files on the Slave Station.

Delay Ticks This field is used to specify a forced delay between export of each record during replication. This forced delay allows other client users to share the time with the server-export process in-between the idled intervals. The value of this delay is specified in ticks (1 tick equalling 1/60th of a second). Any value greater than 60 will be ignored (causing no forced delay). When no time is specified (or specified delay is < 60) no delay is forced and the replication-process (since it is server-side) preempts CPU usage.

Control Parameters In this field the user may enter various parameters separated by a colon (:). These parameters are used to control various aspects of the replication. E.G. Deletelog:UniqueScriptFileName. Currently available commands are shown in table titled “Control Parameters” on page 2.21.5:

Parameter	Function
DeleteLog	Export a delete-action file. This file includes SQL delete statements to delete records as defined in the ERP2020 Delete-Log table. The name of this file is “Dlog_DTStamp.txt”. The Script file will call this file to be executed after all other SQL record-modifications and record-addition commands have been executed. This parameter affects the MYSQL replication only.
UniqueScriptFileName	The script file name by default is “Script.txt”. If this parameters is specified then the script file name is “Script_DTStamp.txt”, where DTStamp is the number that corresponds to the Date-Time-Stamp of the replication export. This parameter affects the MYSQL replication only.

TABLE 1. Control Parameters

Export-File Name-Delimiter Replication data is exported on a table by table basis. The name of each file is created by concatenating the table-number, a delimiter (defined via this field) and the date-time-stamp of the current export. The delimiter must be at least one character and at most 2 characters. The default delimiter is the underscore “_” character.

Accelerated Duty Cycle This field is only available for the record which defines the Master 4D-Export. The value in this field accelerates the 4D-export duty cycle. This is useful in those application where the 4D replication is being

used as a data-backup device, thus requiring a more frequent export than a MYSQL or Oracle export. As an example when the duty-cycle as defined in the HouseKeeping record is set to 2-hours and the value of this field is set to 4, then the 4D replication will be triggered every 30 minutes, while the MYSQL replication (if enabled) will be triggered every 2 hours. The first replication after a server is restarted will always export all requested data, i.e. 4D followed by MYSQL followed by ORACLE assuming all three replications have been enabled. The time of the first replication will be lined up with the Reference Time specified in the HouseKeeping record. Similarly if a replication is forced all replications are generated. (To suppress a particular replication type, void the associated MISCDATA record temporarily). The forced replication will be considered the first accelerated 4D-replication. If the value of this field is "n" then the MYSQL replication will be generated subsequent to the nth 4D-replication, the forced replication being considered as the first.

Time Reference	Server Start-time	Duty Cycle	Acceleration Factor	4D	MYSQL
13:00	12:30	1:00	1	13:00	13:00 (after completion of 4D)
13:00	13:20	1:00	1	14:00	14:00 (after completion of 4D)

TABLE 2.

Ftp Host

Name of the intermediary FTP host.

User Name

User name of the account on the FTP host. Note that the ERP2020 Master Server deposits and receives all files from the default directory of this account. This account must not contain any other files except those deposited by the ERP2020 Master or associated slaves.

When defining an Oracle-Replication Configuration, this user name also doubles as the user-login account into the Oracle data-base. All

tables in the Oracle Database have read/write/delete permissions granted to this user.

Password

Password for the specified account.

When defining an Oracle-Replication Configuration, this password also doubles as the password for the user-login account into the Oracle data-base.

Email Recipient

The email address or the list of email addresses (separated by commas) to whom an email will be sent in case of any major errors encountered during the replication process.

Void

If this field is set then the corresponding replication is suppressed.



Schema

The Schema Button in the top right corner of the Oracle or MYSQL replication configuration allows the ERP2020 administrator to export the ERP2020 Schema for Oracle or MYSQL. Note that for the Oracle schema export all table rights will be granted to the user specified in the User and Password fields above. Schema can be exported only if the configuration record has been properly installed by Semilabs and is current (not expired).

Understanding the Replication Process

To facilitate replication, each table in the ERP2020 has two date-time stamp fields- the Creation Date-time-Stamp field and the Modification Date-Time-Stamp field. The Date-Time-Stamp field is a Long-integer whose value represents the number of seconds since Jan1st 2000. The Modification Date-Time-Stamp field is automatically updated with the date and time on the server when any record in a given table is created or modified. The stamping is done via a table-trigger executed on the server. The Modification Date-time-stamp is updated irrespective of the source of the data-entry or data-modification, i.e. irrespective of whether it is performed via a 4D Client or Web-Browser.

A stored procedure called REPLICATION also runs on the Server Side. This procedure is always on and visible on the server. See Figure 21.3 on page 21.9. The replication procedure exports all data that has been modified since the last export based on the date-time-stamp of the record. The time of the last export (assuming that the entire process has been successful) is saved by the ERP2020 Server in the HOUSEKEEPING RECORD.). There are separate fields for Last-Export Time Stamps for 4D, MYSQL, and ORACLE exports.

The data of changes is first exported to a local directory on the Server Machine. There is one file for each table in the database. The file name is encoded as N_DATETIMESTAMP.TXT (N_DATETIMESTAMP.DAT for Oracle) where N is the table number and DATETIMESTAMP signifies the time of the export. After all tables have been exported to separate files, ERP2020 launches the zip program to compress and integrate the files into a single ZIP file named XXXXXXXX.zip where XXXXXXXX is the value of the datetimestamp of the export (I.E. its beginning datetimestamp).

For the MYSQL export, another file called SCRIPT.TXT is generated. This file contains all the necessary SQL statements to load the exported data into the MYSQL database. The script file assumes that the MYSQL user is loading the exported data-files into a fixed directory named "C:\MYSQL\ZIP". For Oracle data separate Script.CTL files (used by Oracle SQLLDR) are generated for each table. The zipped data-file and Script file (zipped Script files, in case of Oracle) are then transferred to the intermediary FTP site, following which the non-compressed files are deleted from the server's local directory.

When the zipped data-file is transferred to the FTP server, it is first placed there as a ".tmp" file. Immediately thereafter it is renamed as a ".zip" file. This last operation is necessary to check that the transfer indeed did take place and also to make the zip file "invisible" to the slave replication until the transfer has been completed.

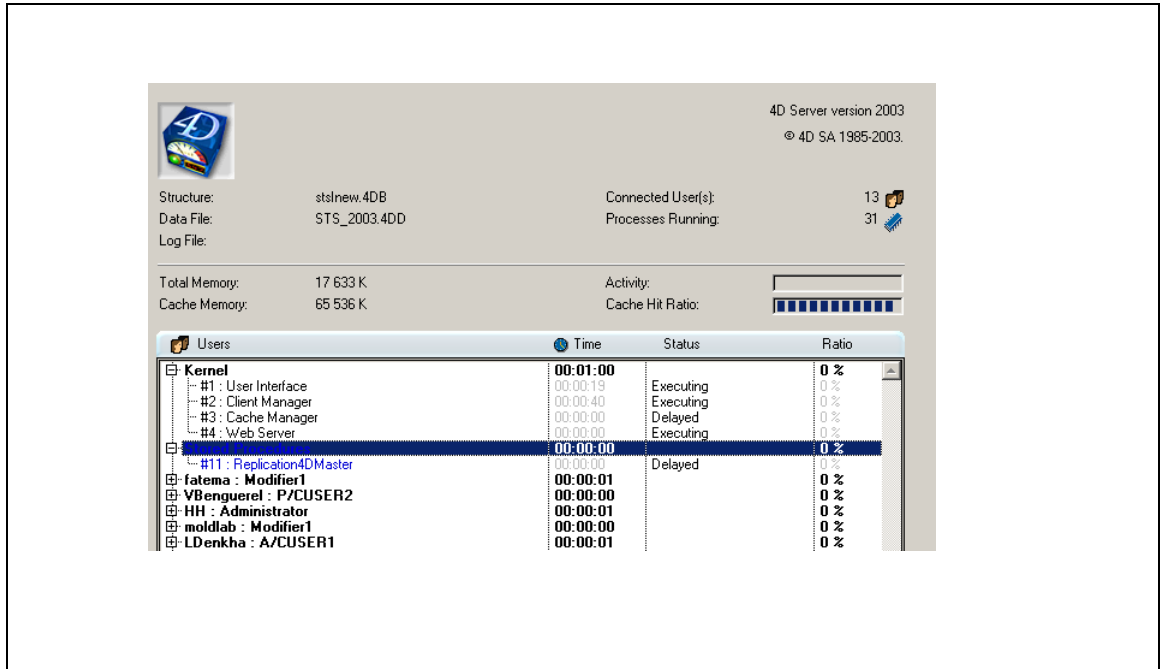


FIGURE 21.3

Replication Timing And Duty Cycle

Once the replication option has been installed and the replication records (in the MISCDATA Table) are properly configured, the export process from the server side is automatic and executes at a user-defined duty-cycle, starting from a base-time. This base-time is stored in the “Replication Time Reference” field of the House Keeping section. For more information on entering the base-time and duty cycle refer to the House Keeping Chapter. See “HouseKeeping-record fields and check boxes :” on page 9.13. After a successful export, the House Keeping record is updated to store the date & time of the last export in the “Last-Export Time Stamp” field. This value is then used as the threshold to determine the incremental changes for the next replication. See also note on “Accelerated Duty Cycle” under “” on page 2.21.4

Resetting The Beginning Date-time Stamp

The “Last-Export Time Stamp” field in the HOUSEKEEPING RECORD may be modified if the selection criteria (based on time) for modified data (to be exported) needs to be changed. For example, to export the entire database the Last-Export Time Stamp may be set to 0 (zero). This feature can also be used to re-load older data into the slave database if the data on the slave-data needs to be re-created due to a data-loss. The beginning datetimestamp should be calculated and entered into the Housekeeping record. Note the Last-Export Time Stamps are stored separately for 4D, MYSQL, and ORACLE exports.

Under normal circumstances the date-time-stamp will be set backwards to request older data that has been missed. If the user sets the value forwards, an error will be issued and the value reverts to the previously saved value. The system however does provide the user the option to set the time forward. This can be done if after changing the value, the user presses the Windows-Control-Key before the user exits the field-area by pressing TAB or by clicking on another field.

Starting Replication

The replication-process on the server side is launched automatically upon server startup and delayed by the time defined in the House Keeping Record. This automatic launch may be suppressed by pressing the WINDOWS and CTRL Keys when the server is being started. Note that in a single-user application the reverse is true. The Replication Process does not start automatically. The Windows and CTRL Keys must be pressed when the password dialog appears and must be kept down until the Replication process is started

Updating of the HouseKeeping Record

As stated earlier, if the entire operation has been successful, the server updates the HOUSEKEEPING record to store the DATETIMESTAMP of the starting time-marker of the NEXT export. To ensure there can be no data missed, this time stamp is set back by either two hours or 5 minutes, depending on the setting in the Admin-Control field of the HouseKeeping record. (See “Date-Time stamp set-back on ERP2020 master” on page 9.18.) A subsequent replication-export will have an overlap of data with the prior export. The amount of overlap is determined by the set-back of the DATETIMESTAMP stored in the Housekeeping record.

Special characters and formats in exported data

During the MYSQL export a backslash “\” character will be replaced by the following string of characters: “\\”. It is the responsibility of the MYSQL server to re-convert the string of characters back to a back-slash.

When text data is being exported (as opposed to strings), carriage-returns (ASCII-Code 13) and TABS (ASCII-Code 9) are also replaced with a space-character (ASCII-Code 32). Text data can be up to 32,000 characters long.

A null-date in the ERP2020 (00/00/00) will be sent as a null (nothing between two tabs).

Record-deletion log

There are other considerations that come into play in the replication process resulting from the deletion of records from the ERP2020 database. A separate table called DELETEDLOG contains the table-number and Primary-Key field of each record that has been deleted from the ERP2020 Master database. Records deleted during the normal archiving process are not recorded in the DELETEDLOG Table.

Entries in the Delete-Log table are trimmed when the “Data Update Procedure” is run. The trimming process deletes all entries in the Delete-Log table whose creation date-time stamp is less than the date-time stamp of the earliest of the of successful replication exports of MYSQL, 4D and/or Oracle. The date-time stamps of the successful replication exports of MYSQL, 4D and Oracle are saved in the Housekeeping record of the ERP2020 Master database.

Replication behavior during the Archiving Process

Note that the archiving process has two phases, each phase being executed at the user's option. In Phase 1 all closed Jobs (records in the RECEIVER Table) are transferred into a different (ARCHIVES) table in the database and then **deleted**. The related LOTINFO and LOTSTEPS and MATERIAL-SUSED records are kept in their place and stamped with the month and year of the Archive Phase 1 process. As to whether or not, the ERP2020 master will create records in the DELETEDLOG table (when records in the RECEIVER table are deleted, as stated above), depends upon the setting in the Admin-Control field of the HouseKeeping record. See “Job deletion from Slave during level-1 Archiving” on page 1.9.19.

During Phase 2 of archiving, some of the records are removed from the database (transferred into a separate data-file). This is done purely to minimize the size (and thereby maximize the speed) of the active data-file. These records are removed when they are deemed “old” and therefore dispensable to the active database. This last process does **not** generate any records in the DELETEDLOG table.

Slave Server’s handling of Archiving

The 4D Slave Server is setup to respond to Phase 1 archiving. It automatically deletes records from the RECEIVER table when a job has been archived (upon detection of a new record added in the

ARCHIVES table). It does not respond to Phase 2 archiving. The 4D Slave Server also does not create any records in DELETEDLOG table if any records are deleted from the Slave database.

In the case of the MYSQL and Oracle Slave Servers it is the responsibility of the MYSQL and ORACLE database administrators to take the necessary action within the Slave databases to respond to level-ii archiving in the Master ERP2020 database. The deletion of records in the Slave-database during level-1 archiving can be automatically handled by the Master depending upon the Admin-Configuration-Control in the HouseKeeping record.

Note:



Starting with Revision I of the ERP2020 (2003 version), the Master-replication-export (upon the user's election, see DeleteLog 21.5) will also include an SQL command file that includes SQL statements to delete the records defined in the DELETEDLOG table. The deletion is time-sensitive, i.e. only those records are deleted whose modification date-time stamp is less than or equal to the creation date-time stamp of the delete-record that dictates the deletion.

Replication Flow-Chart. (see figure 21.4 on page 21.13 and figure 21.5 on page 21.14)

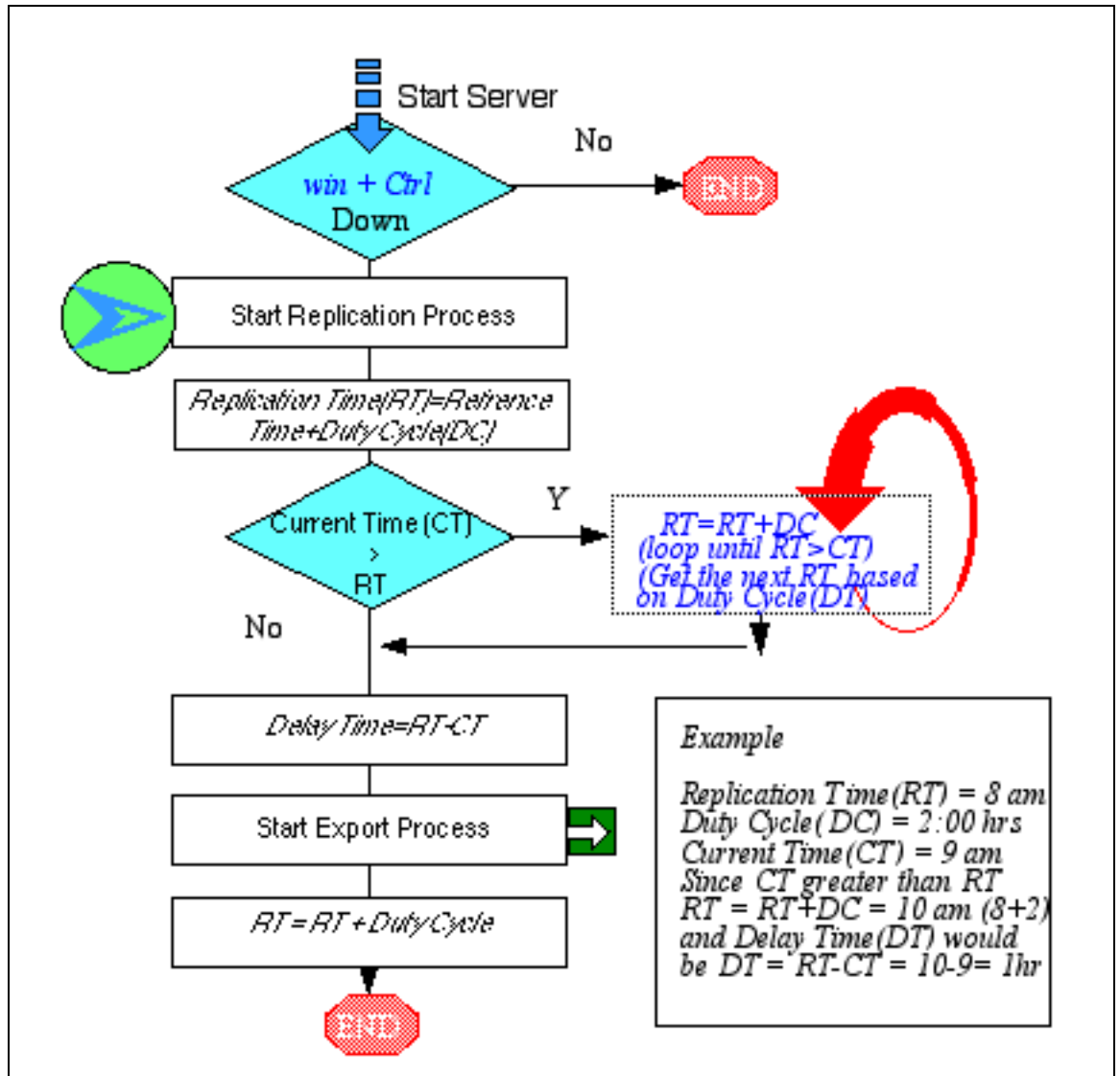


FIGURE 21.4

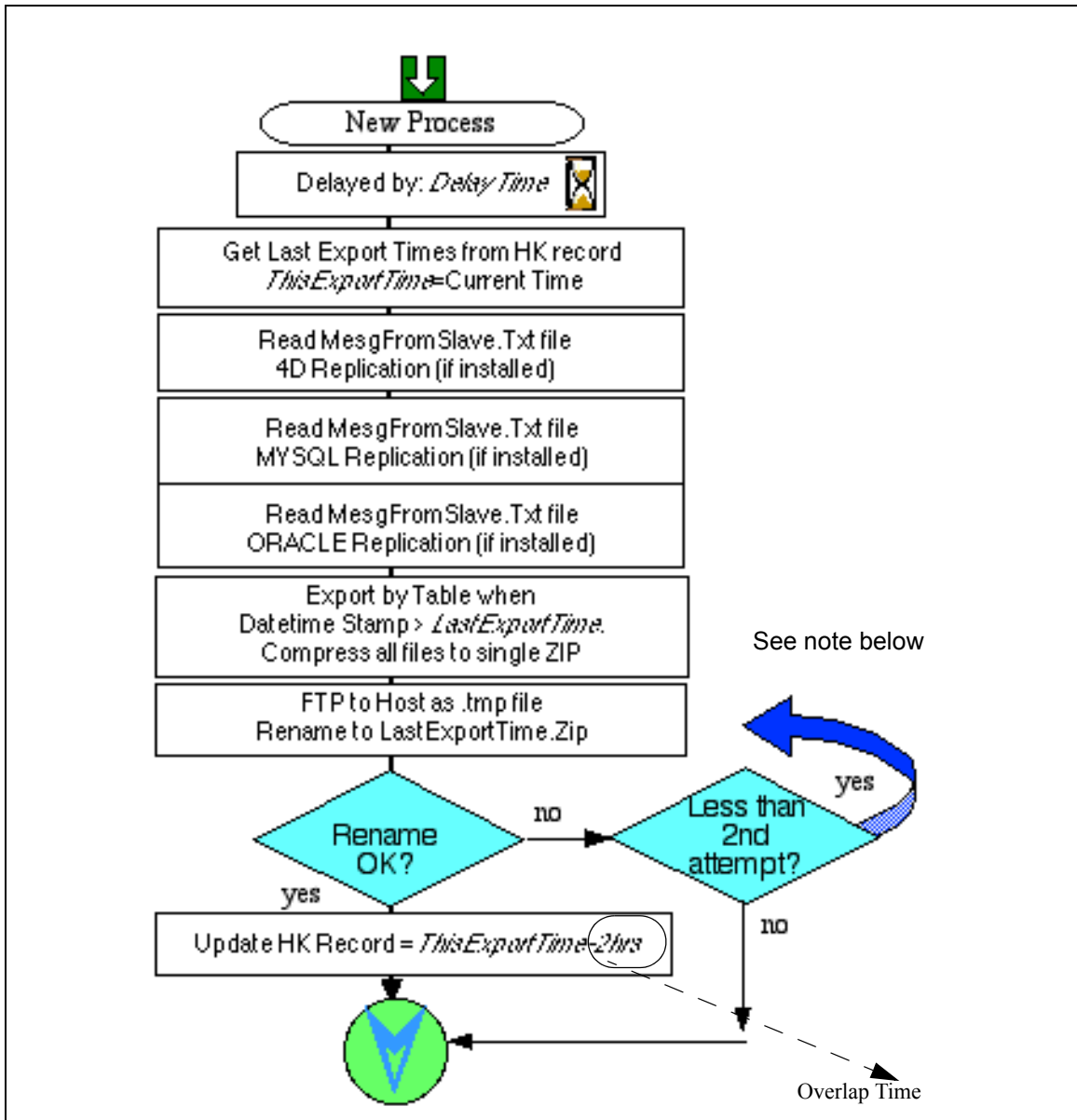


FIGURE 21.5

Replication-Process Notes:

Special consideration for Zero-Date-time-stamp export and import

When the ERP2020 Master server is exporting data for time-stamp ZERO (all records in the database) the Master Server will delete all files that exist in the FTP-Host repository folders. This is to make sure that the remote Slave Replication does not erroneously use older files albeit with a Date-Time Stamp greater than zero. On the Slave side if it is a 4D Slave, it will delete **all** records in each table **before** importing the data from the time-stamp ZERO export. No additional deletion (based on the records received in the Delete-Log table) will take place on the ERP2020 4D Slave server.

Creating a ZIP file

After the ERP2020 Master-Server has exported all the relevant records in all the tables, it launches the compression process by invoking the zip-utility via the "Run" (DOS) command. The compression process thus launched runs asynchronously from the ERP2020 process. The Master-server therefore performs a periodic check (via a loop) to test the creation of the ZIP file. The loop is maintained until the ZIP file has been created (and found by the ERP2020) or the loop has timed-out. The default time-out value of the loop is 300 seconds, however if the sum of the files sizes of all the pre-compression-files is greater than 30 mega-bytes, the delay is set to 10-seconds per mega-byte to be zipped. During the loop the Server process is idled so that server-bandwidth is not compromised. The idle-time within the loop is 10 seconds when less than 30 mega-bytes of files need to be compressed. When the pre-compression sizes of the files exceeds 30 mega-bytes, the idle-time is set to 30-seconds.

Replication overlap time

When a replication is successfully completed, the time-threshold for the next replication is set back by two-hours (default condition). The setting back of this time provides an overlap of data for the next replication. This overlap-time may be changed 5 minutes by setting bit-17 in the Admin-Control field of the HouseKeeping record. See "Date-Time stamp set-back on ERP2020 master" on page 9.18.

Creating a Slave-Server

The ERP2020 Slave-Server is replicated from the ERP2020 Master-server. To install an ERP2020 slave-server perform the following steps:

- Use a backup of the Master Server's data file and install on another Server. Use an identical copy of the ERP2020 to open the data-base.
- Add a [MISCDATA] record named "ReplicationDataSlave4D" in the Slave-Database. If this record already existed on the Master then make this record active.
- Add web-menu (method Web_SlaveServerStatus) for Slave-Status (if one does not exist).
- Change the HouseKeeping record to configure the data-base as a Slave. See "Station Code" on page 9.13
- Add Master's URL in Housekeeping record to enable SOAP connection to Master. See Master/Slave URL on page 9.14
- Restart Server and make sure the stored procedure "**Replication4DSlave**" is on
- Change Date-Time-Stamp on Master to set the time-threshold for the next 4D export of the Master

Converting a Slave to a Master.

- Change the Housekeeping record to configure as a Master. See "Station Code" on page 9.13
- Restart data-base.



Notes: The Housekeeping record does not get replicated between the Master and the Slave servers. When switching between Slave and Master the user must make sure that the House-keeping configuration is completely accurate. This process is done manually.

Replication-Control From Slave-Side

The ERP2020 system provides some limited replication-control capabilities on the Slave Side. The Slave controls are instituted by having the slave place encoded messages in a file called MESSAGESFROMSLAVE.TXT. This file must be placed in the intermediary FTP site. Before starting the replication export process, the Master Server checks for the existence of this file (see flow chart in figure 21.5 on page 21.14). If the file exists it is fetched by the Master Server and then deleted from the FTP site. Any action taken, based on the encoded messages in the file apply to current replication only. During the next replication cycle (if a new MESSAGESFROMSLAVE.TXT) has not been deposited, the ERP2020 replicates in the default mode without any modified actions. It is the responsibility of the Slave to ensure that the Master has already read and therefore deleted a previous message file before a new one can be put in the FTP site.

The Slave may change the DateTimeStamp (used to determine the date& time filter for the data to be exported).

RESET_DATETIMESTAMP mode nn Where “mode” is either “4D”, “MYSQL”, or “ORACLE” and “nn” is the new date-time stamp for the next 4D, MYSQL, or ORACLE (depending on mode) replication export. Note that the command and the operand are separated by a “space” which acts as the delimiter. The “.” and “#” characters may also be used as delimiters. NOTE: On the 4D Slave Server if the user changes the Date Time Stamp of the last import the Slave Server will (at the user’s option) automatically create the MESSAGESFROMSLAVE.TXT with the user specified date-time stamp and FTP it to the FTP account so that the new export from the master includes records with the new date-time stamp requested by the slave.

CHOOSE_TABLE mode T1 T2 This command will provide a table-specific export, i.e. data will be exported from only the table(s) whose number is listed via “T1”, “T2” etc. (Table numbers must be separated by space as shown in the command syntax on the left).

Note i) During the table-specific export the ERP2020 will still generate empty files for tables which were not selected via this command. This is necessary because the ERP2020 looks for files for all tables before the zipping can proceed. During a table-specific export the date-time-stamp in the Housekeeping record is NOT updated after the export so that a subsequent export of all files does not miss any data. Also

when a user requests a table-specific export with a date-time stamp of zero, the date-time stamp is changed to 1. This change is made because on the ERP2020 4D slave station a date-time stamp of zero forces a delete of all records in a table prior to the import.

MESGFROMSLAVE.TXT Examples:

{Example 1: MYSQL export for table 2 only. Date-time stamp of 51783110}

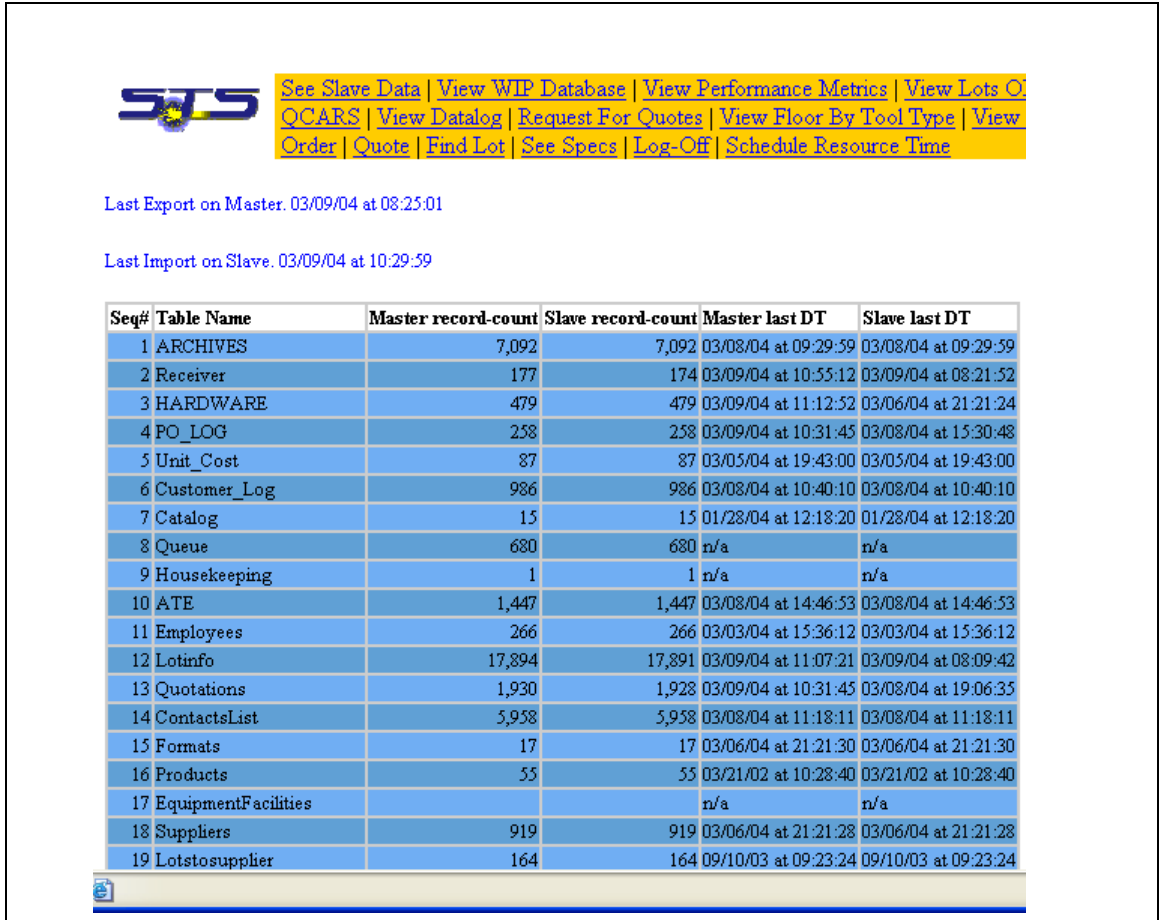
```
RESET_DATETIMESTAMP MYSQL 51783110  
CHOOSE_TABLE MYSQL 2
```

{Example 2: 4D export for table 12 only. Date-time stamp of 0}

```
RESET_DATETIMESTAMP 4D 0  
CHOOSE_TABLE 4D 12
```

Comparing the Slave-Data to the Master

The ERP2020 Slave Server allows the administrator to create a snap-shot of the data on the Slave and Master. This facility is invoked via a web-browser connected to the Slave. When the browser requests the status, the Slave-Server compiles a list of all records in all the tables of the Slave and the time of the last update of each table. It then makes a SOAP request to the Master-Server and collects the same information. The information from the master is integrated with the information on the Slave and presented to the user in the same form Figure 21.6 on page 21.19. Note that for the Slave to be able to make a connection to the Master Server, the IP address of the Slave must be defined in the HousKeeping record of the Master-Server.



Seq#	Table Name	Master record-count	Slave record-count	Master last DT	Slave last DT
1	ARCHIVES	7,092	7,092	03/08/04 at 09:29:59	03/08/04 at 09:29:59
2	Receiver	177	174	03/09/04 at 10:55:12	03/09/04 at 08:21:52
3	HARDWARE	479	479	03/09/04 at 11:12:52	03/06/04 at 21:21:24
4	PO_LOG	258	258	03/09/04 at 10:31:45	03/08/04 at 15:30:48
5	Unit_Cost	87	87	03/05/04 at 19:43:00	03/05/04 at 19:43:00
6	Customer_Log	986	986	03/08/04 at 10:40:10	03/08/04 at 10:40:10
7	Catalog	15	15	01/28/04 at 12:18:20	01/28/04 at 12:18:20
8	Queue	680	680	n/a	n/a
9	Housekeeping	1	1	n/a	n/a
10	ATE	1,447	1,447	03/08/04 at 14:46:53	03/08/04 at 14:46:53
11	Employees	266	266	03/03/04 at 15:36:12	03/03/04 at 15:36:12
12	Lotinfo	17,894	17,891	03/09/04 at 11:07:21	03/09/04 at 08:09:42
13	Quotations	1,930	1,928	03/09/04 at 10:31:45	03/08/04 at 19:06:35
14	ContactsList	5,958	5,958	03/08/04 at 11:18:11	03/08/04 at 11:18:11
15	Formats	17	17	03/06/04 at 21:21:30	03/06/04 at 21:21:30
16	Products	55	55	03/21/02 at 10:28:40	03/21/02 at 10:28:40
17	EquipmentFacilities			n/a	n/a
18	Suppliers	919	919	03/06/04 at 21:21:28	03/06/04 at 21:21:28
19	Lotstosupplier	164	164	09/10/03 at 09:23:24	09/10/03 at 09:23:24

FIGURE 21.6

The right-most column of the display shows the status/comments related to the table. This column may also show an error condition, if one exists. The following error conditions are checked:

Error

If the date-time-stamp of the last modification of a table on the Master and the Slave is the same and if the record-count in the Slave-table is **less** than the record-count of the Master-table then it indicates missing records on the Slave. This error condition is flagged. If the user is

logged in as the designer then the error-message also includes a link that provides a listing of records (via the Primary-Key) that exist on the master and are missing on the Slave. When record comparisons are made the Prmiary Keys from the Master table are fetched into the slave in maximum sections of 100,000 records.

ERP2020 4D Slave Server

The ERP2020 Slave Server imports files exported by the master server and updates the slave data-base with the new modifications. The flow diagram is shown in figure 21.7 on page 21.21.

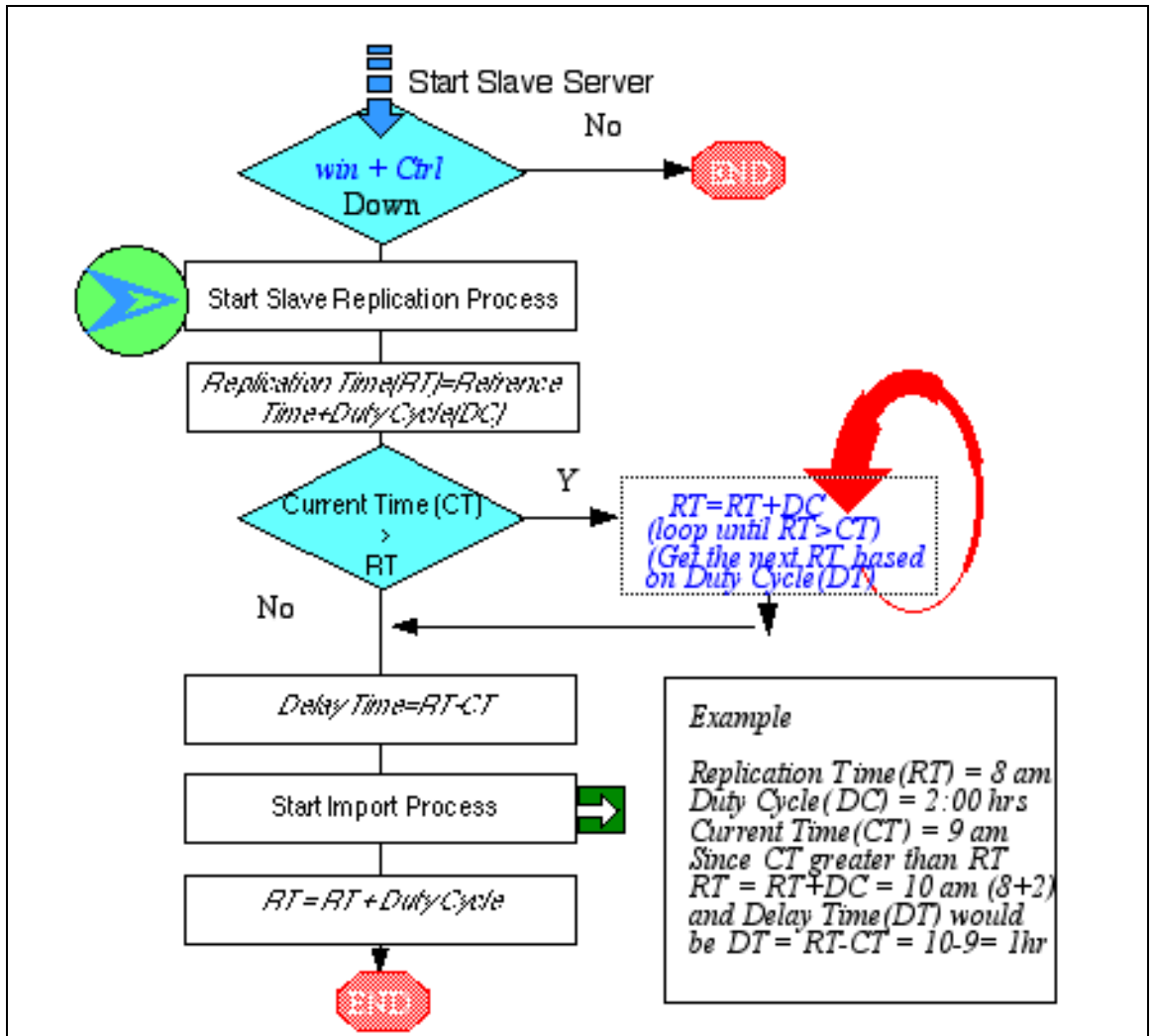
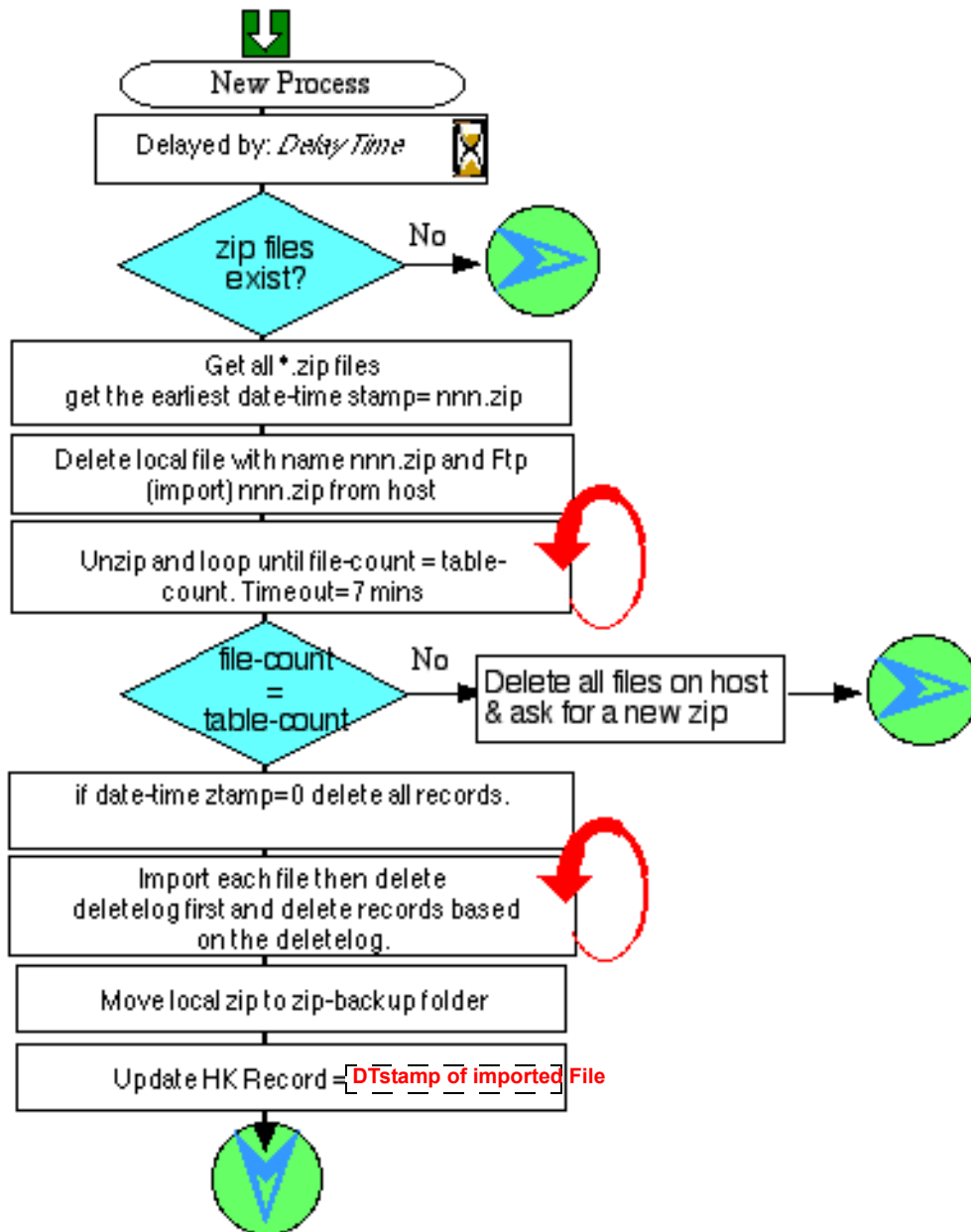


FIGURE 21.7



Notes:

Uncompressing a ZIP file

After the ERP2020 Slave-Server has exported the earliest ZIP file from the FTP-host, it launches the de-compression process by invoking the zip-utility via the "Run" (DOS) command. The de-compression process thus launched runs asynchronously from the ERP2020 process. The Master-server therefore performs a periodic check (via a loop) to monitor the de-compression process. The loop is maintained until the files have been extracted or the loop-time-out has expired, which-ever occurs first. The loop-time-out is set to 20-seconds times the size of the ZIP file in mega-bytes. If the ZIP file-size is less than 1 mega-byte then the time-out is set to the default of 5 minutes.

4D Slave Error Processing:

If the ERP2020 4DSlave Server encounters an exception a mail record is created inside the ERP2020 data base, detailing the error condition. Typical error conditions include loss of communication with the FTP server, no ZIP file on the FTP server or a corrupted ZIP file on the FTP server.

If the ZIP file from the Master Server is corrupted the 4DServer Slave will delete all ZIP files present on the FTP site and request (via the MESSAGESFROMSLAVE.TXT file) for a re-send of the replication with the same Date-Time stamp as of the corrupted ZIP file. The master server (after reading the MESSAGESFROMSLAVE.TXT file) will send the next replication which includes all data modifications made since the requested date-time stamp. This constitutes a self-correcting loop requiring no user intervention.

Table Triggers

Table triggers update the creation and modification datetimestamp fields for a record, when the record is first created or later modified. When a record is deleted the trigger results in the creation of a record in the Delete_log table. The table below shows the table-names and triggers associated with them. A "Create", "Modify" or "Delete" tagged with the table-name indicates that a trigger is launched when the corresponding action takes place.

Table	Triggers		
ARCHIVES	CREATE	MODIFY	
RECEIVER	CREATE	MODIFY	DELETE
HARDWARE	CREATE	MODIFY	DELETE
PO_LOG	CREATE	MODIFY	DELETE
UNIT_COST	CREATE	MODIFY	DELETE
CUSTOMER_LOG	CREATE	MODIFY	DELETE
CATALOG	CREATE	MODIFY	DELETE
QUEUE			
HOUSEKEEPING	CREATE	MODIFY	DELETE
ATE	CREATE	MODIFY	DELETE
EMPLOYEES	CREATE	MODIFY	DELETE
LOTINFO	CREATE	MODIFY	DELETE
QUOTATIONS	CREATE	MODIFY	DELETE
CONTACTSLIST	CREATE	MODIFY	DELETE
FORMATS	CREATE	MODIFY	DELETE
PRODUCTS	CREATE	MODIFY	DELETE
EQUIPMENTFACILITIES	CREATE	MODIFY	DELETE
SUPPLIERS	CREATE	MODIFY	DELETE
LOTSTOSUPPLIER	CREATE	MODIFY	DELETE
QCARS	CREATE	MODIFY	DELETE
DOC_CONTROL	CREATE	MODIFY	DELETE
LOTSTEPS	CREATE	MODIFY	DELETE
MAIL_MESSAGES			
MAIL_USERS			
REPAIR_LOG	CREATE	MODIFY	DELETE
BUY_ORDERS	CREATE	MODIFY	DELETE
BUY_ITEMS	CREATE	MODIFY	DELETE
AVL_SUPPLIES	CREATE	MODIFY	DELETE
CHECK_REGISTER	CREATE	MODIFY	DELETE
BO_APPROVERS	CREATE	MODIFY	DELETE
CHART_OF_AC	CREATE	MODIFY	DELETE

ASSET_LIST	CREATE	MODIFY	DELETE
SAVE_LISTS	CREATE	MODIFY	
RECEIVABLES	CREATE	MODIFY	DELETE
DEPOSITS	CREATE	MODIFY	DELETE
STEPS_FILES	CREATE	MODIFY	DELETE
STEPS_TEMPS	CREATE	MODIFY	DELETE
HELP	CREATE	MODIFY	DELETE
LOG_BOOK	CREATE	MODIFY	DELETE
CAL_EVENTS	CREATE	MODIFY	DELETE
CREDIT_MEMO	CREATE	MODIFY	DELETE
CASH	CREATE	MODIFY	DELETE
USER_ATTRIBUTES	CREATE	MODIFY	DELETE
PICTS	CREATE	MODIFY	DELETE
USER_TEMPLATES	CREATE	MODIFY	DELETE
WWWDILOGS	CREATE	MODIFY	DELETE
OLDPOS	CREATE	MODIFY	DELETE
OLE	CREATE	MODIFY	DELETE
TEMPLATE_DEFINITIONS	CREATE	MODIFY	DELETE
CUSTOMER_SALES	CREATE	MODIFY	DELETE
INVENTORY	CREATE	MODIFY	DELETE
BUILD_CARD	CREATE	MODIFY	DELETE
LABELS			
ERP_HELP	CREATE	MODIFY	DELETE
INV_USAGE_MANY	CREATE	MODIFY	DELETE
RESOURCE_CALENDAR	CREATE	MODIFY	DELETE
DELETELOG			
DEPOSIT_ITEMS	CREATE	MODIFY	DELETE
CM_ITEMS	CREATE	MODIFY	DELETE
BOM	CREATE	MODIFY	DELETE
BOM_ITEMS	CREATE	MODIFY	DELETE
EMPLOYEECERTDEF	CREATE	MODIFY	DELETE
MISCDATA	CREATE	MODIFY	DELETE
RM_REPORTS	CREATE	MODIFY	DELETE
LOTCONTAINERS	CREATE	MODIFY	DELETE
MATERIALSUSED	CREATE	MODIFY	DELETE
PO_ITEMS	CREATE	MODIFY	DELETE
BO_APPROVERS2	CREATE	MODIFY	DELETE
PARTDATA	CREATE	MODIFY	DELETE
WEB_MENUS	CREATE	MODIFY	DELETE
CURRENCYDATA			
WEB_USER_SETTINGS			

DEBUGLOG			
BINNINGCONTROL	CREATE	MODIFY	DELETE
CODESTABLE	CREATE	MODIFY	DELETE
AREADEFINITION	CREATE	MODIFY	DELETE
CONSOLIDATEDSHIPPER	CREATE	MODIFY	DELETE
FORECASTDATA	CREATE	MODIFY	
CUSTOMERRMAS	CREATE	MODIFY	DELETE
RESERVATIONS	CREATE	MODIFY	DELETE
ACCTTRANSACTION	CREATE	MODIFY	DELETE
MONTHLYGLBALANCE	CREATE	MODIFY	DELETE
ACCOUNTINGCONSTANTS			
MENUCONTROL	CREATE	MODIFY	DELETE
DEVICETABLE	CREATE	MODIFY	DELETE
CURRENCYQUOTES	CREATE	MODIFY	DELETE
SALESTAXTABLE	CREATE	MODIFY	DELETE
QCARTeam	CREATE	MODIFY	DELETE
SLAVEDRIVENCHANGES			
TRIGGERMESSAGE	CREATE		

NOTES:2) TRIGGERS ARE NOT EXECUTED ON A 4D SLAVE SERVER DURING THE REPLICATION-IMPORT PROCESS.

SOAP-Connectivity between Master and Slave

Changing Data on the Slave & Updating it on the Master:

Starting with revision H9 of the ERP2020, data-modified on a Slave Server may be updated on the Master Server. This feature is useful for installations, where access is not directly available to the Master Server because the Master is behind a fire-wall. This feature may also be desirable when the 4D-Client connection to a Master (located outside the local-area network) is sluggish or intermittent due to a high-latency internet-connection. Note that this data interchange is not the same as the replication process discussed in the preceding sections. The replication process is a batch-process in which the Slave-Server updates itself periodically from batch-data that is deposited by the Master-Server (periodically) on a common FTP-host.

The mode of data-interchange between the Master and Slave is dependent upon the availability or non-availability of the Master and the Slave outside the fire-wall. There are three possible cases;

Case1)

Master & Server have access to each other. They are both outside the fire-wall. In this case any changes made to the Slave will cause the Slave to execute a WEB-Services call to the Master. This call is made from inside the record-modification table-trigger on the Slave Server. For this call to work properly the Housekeeping record on the Slave Server must be configured as follows:

- a) The SOAP interface bit (in the Designer-Control field) must be set. See “Automatic WEB-Services call by Slave to Master” on page 9.21.
- b) The Server should be configured as a Slave. See “HouseKeeping-record fields and check boxes :” on page 9.13.
- c) The URL of the Master Server must be defined. See “HouseKeeping-record fields and check boxes :” on page 9.13. The URL must have the complete IP address with the port-number on which the ERP2020 web-server is being published.
- c) The Remote-User -Name and Remote-User-Password fields must contain valid data. This User must exist in the Users and Groups of the Master-Server. See “HouseKeeping-record fields and check boxes :” on page 9.13.

Case2)

The Master is outside the fire-wall and the Slave is inside a fire-wall but has access to the Master. The connection mode in this case is similar to the preceding case, because the Slave initiates the WEB-Services call.

Case3)

The Slave is outside the fire-wall and the Master is inside the fire-wall but has access to the Slave. In this case the Slave cannot make a WEB-Services call to the Master, therefore a Client connected to the Master must manually request for a data transfer from the Server to the Master. The Master network (via a Client, as opposed to the Master-Server itself) initiates the WEB-Services call.

Slave Server Changes Rules:

1) The Slave-Server allows records to be **modified** but does **not** allow records to be **created**. When any records are modified on the Slave-Server, copies of the modified records are saved in a separate table called "SlaveDrivenChanges". The modified records contain only the direct user-modifications without any alterations to the Creation and Modification Date-time-Stamp fields that were set by the Master. These fields are only updated when a record is created or modified on the Master Server.

2) When changes are made on the Slave-Server, some of the changes are made in a transaction. A transaction is necessary when changes affect more than one table. The data-update must happen in all the tables or none of the tables. For example when updating a buy-item, the buy-order may also be updated and changes to both the tables must be saved together (or cancelled) as one transaction. To make sure that the updates on the Master-Server are also processed in a transaction, the Slave-Server binds the changes in the transaction with a common Transaction-ID and communicates this information to the Master-Server. The Master-Server will update the data if all tables affected by the transformation will allow the update.

3) If the Slave-Server is setup to automatically update the Master, it must be configured appropriately in the HouseKeeping record. See "Automatic WEB-Services call by Slave to Master" on page 9.21. The HouseKeeping record on the Slave-Server must also contain information that identifies the Master URL, user-name and password.

4) In the case when the Slave will **not** automatically update the Master but the Master will make a specific call to the Slave, the Slave-Server must have a user-account setup so that it can authenticate a Client (logged into the Master-Server) requesting a SOAP-transfer of data.

Master Server data-update Rules:

Update on the Master Server may be automatic or based on a request made from a Client connected to the Master. In the automatic mode, any changes made on the Slave are automatically sent to the Master via Slave-Server triggers (assuming the Slave-Server is configured (via its HouseKeeping record) to perform the trigger-based SOAP call to the Master. Correspondingly, the Master must be accessible via the Slave-Server and must authenticate the Slaver-Server call by verifying the user-name and password passed by the Slave to the Master.

1) The update on the Master-Server is allowed if the following conditions are met:

- a) The record being updated exists on the Master-Server
- b) The Modification date-time stamp of the record(s) on the Master-Server matches the Modification date-time stamp of the record(s) being downloaded from the Slave-Server. This condition ensures that the record modified on the Slave-Server was the latest record and was not modified on the Master since the time of the last replication on the Slave

2) In the mode wherein the Slave has no direct access to the Master (Case 3 above), the data-update on the Master is performed when a **Client** logged into the Master-Server (as Administrator) specifically requests the update from the Slave-Server. The request is made via the **Housekeeping**-Menu-Bar, -- Menu: **Remote Servers**,-- Menu-item: **Get from ERP2020 Slave Server**. The actual request is executed via a WEB-Services call to the Slave-Server. The Slave-Server must therefore be accessible to the Client (logged into the Master-Server) via the internet.

3) When the Client (logged into the Master-Server) requests data from Slave-server a dialog requesting the Slave-Server URL, User-Name and Password is presented. The User-Name and Password must exist and match with the User-Name and Password **stored on the Slave-Server**. This authentication must be cleared before the data from the Slave-Server will be updated. The interface dialog is shown below in Figure 21.8 on page 21.31

•Menu: Remote Servers**Menu-Item:** *Get From ERP2020 Slave Server*

This menu-item allows a Client logged into the Master-Server to make a web-services call to the Slave-Server to fetch all modifications made on the Slave Network and then incorporate them into the Master-Server. This menu-item is used when the Slave-Server cannot make an automatic call to the Master-Server (for example in the case when the Master-Server is inside a fire-wall, and therefore not accessible to the Slave-Server) to update the Master-Server with changes on the Slave-Side. The Client is a WEB-Services Client to the Slave-Server.

Menu-Item: *Update ERP2020 Master Server*

This menu-item allows a Client logged into the Slave-Server to make a web-services call to the Master-Server to deposit all modifications made on the Slave Network to the Master-Server. This menu-item is used when the Slave-Server cannot make an automatic call to the Master-Server. The Client on the Slave-Side is a WEB-Services Client to the Master-Server.

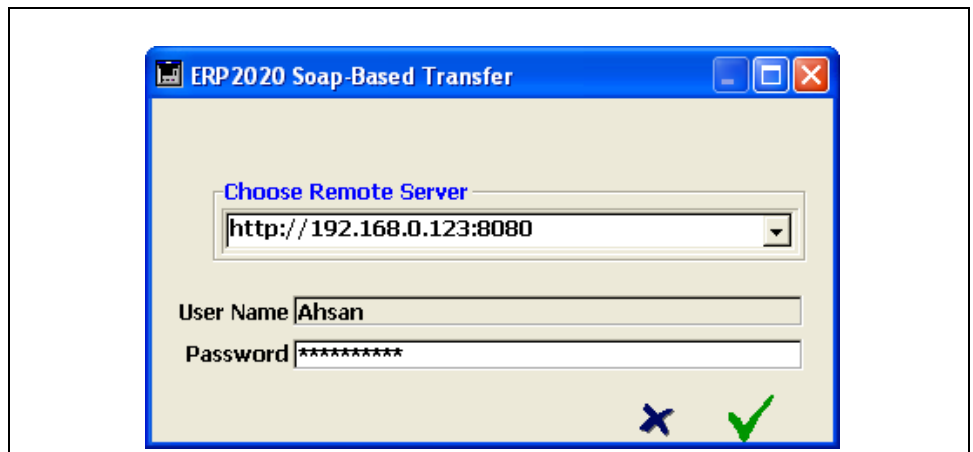


FIGURE 21.8

Remote Server URL

The remoter-server URL is chosen from a pull-down menu. This menu is populated via one of two the Hierarchical Lists called

“SlaveServerURLs” and “MasterServerURLs”. If the current user is logged into a Slave-based-network then the “MasterServerURLs” list is used. If the current user is logged into a Master-based-network then the “SlaveServerURLs” list is used.

User Name

If the current user is logged into a Slave-based-network then the user-name entered here must exist on the Users and Groups defined in the Master Server. If the current user is logged into a Master-based-network then the user-name entered here must exist on the Users and Groups defined in the Slave Server.

Password

If the current user is logged into a Slave-based-network then the password entered here must exist on the Users and Groups defined in the Master Server. It will be validated on the Master. If the current user is logged into a Master-based-network then the password entered here must exist on the Users and Groups defined in the Slave Server. It will be validated on the Slave.

Primary-Keys by Table

1	ARCHIVESRECEIVER_NUMBER {8}	TYPE:L	INDEXED & UNIQUE
2	RECEIVERRECEIVER_NUMBER {8}	TYPE:L	INDEXED & UNIQUE
3	HARDWAREUNIQUEID {60}	TYPE:A	INDEXED
4	PO_LOGPO_NUMBER {4}	TYPE:A	INDEXED & UNIQUE
5	UNIT_COSTUNIQUEID {20}	TYPE:A	INDEXED
6	CUSTOMER_LOGCUST_CODE {10}	TYPE:A	INDEXED & UNIQUE
7	CATALOGITEM {1}	TYPE:A	INDEXED
8	QUEUELOTNUM {1}	TYPE:A	INDEXED
9	HOUSEKEEPING		
10	ATE_UNIQUEID {51}	TYPE:A	INDEXED & UNIQUE
11	EMPLOYEESEMPLOYEE_CODE {9}	TYPE:A	INDEXED & UNIQUE
12	LOTINFOID {37}	TYPE:A	INDEXED & UNIQUE
13	QUOTATIONSQUOTENUMBER {1}	TYPE:A	INDEXED & UNIQUE
14	CONTACTSLISTCONTACTCODE {1}	TYPE:L	INDEXED & UNIQUE
15	FORMATSUNIQUEID {6}	TYPE:A	
16	PRODUCTSUNIQUEID {14}	TYPE:A	INDEXED & UNIQUE
17	EQUIPMENTFACILITIESNAME {1}	TYPE:A	
18	SUPPLIERSUNIQUEID {40}	TYPE:A	INDEXED
19	LOTSTOSUPPLIERID {12}	TYPE:A	INDEXED
20	QCARSQCARNUMBER {1}	TYPE:L	INDEXED & UNIQUE
21	DOC_CONTROLUNIQUEID {14}	TYPE:A	INDEXED
22	LOTSTEPSSEQ_NUMBER {108}	TYPE:L	INDEXED & UNIQUE
23	MAIL_MESSAGES		
24	MAIL_USERSUSER_NAME {1}	TYPE:A	INDEXED & UNIQUE
25	REPAIR_LOGREP_NUM {2}	TYPE:L	INDEXED & UNIQUE
26	BUY_ORDERSSEQ_NUM {1}	TYPE:L	INDEXED & UNIQUE
27	BUY_ITEMSSEQ_NUMBER {55}	TYPE:L	INDEXED
28	AVL_SUPPLIESUNIQUEID {10}	TYPE:A	INDEXED & UNIQUE
29	CHECK_REGISTERCHECK_NUM {1}	TYPE:L	INDEXED & UNIQUE
30	BO_APPROVERSUNIQUEID {8}	TYPE:A	INDEXED
31	CHART_OF_ACUNIQUEID {19}	TYPE:A	INDEXED
32	ASSET_LISTASSET_NUM {1}	TYPE:L	INDEXED & UNIQUE
33	SAVE_LISTSLIST_NAME {1}	TYPE:A	INDEXED & UNIQUE
34	RECEIVABLESUNIQUEID {23}	TYPE:A	INDEXED
35	DEPOSITSSLIPNUM {1}	TYPE:L	INDEXED & UNIQUE
36	STEPS_FILESUNIQUEID {13}	TYPE:A	INDEXED
37	STEPS_TEMPUNIQUEID {6}	TYPE:A	INDEXED
38	HELP_TOPIC {3}	TYPE:A	INDEXED & UNIQUE
39	LOG_BOOKPAGE {2}	TYPE:L	INDEXED & UNIQUE
40	CAL_EVENTS		
41	CREDIT_MEMOCMNUM {1}	TYPE:L	INDEXED & UNIQUE
42	CASHAC_NUM {3}	TYPE:A	INDEXED & UNIQUE
43	USER_ATTRIBUTESUNIQUEID {11}	TYPE:A	INDEXED
44	PICTSUNIQUEID {6}	TYPE:A	
45	USER_TEMPLATESUNIQUEID {22}	TYPE:A	
46	WWWDIAGLOGS		
47	OLDPOSUNIQUEID {19}	TYPE:A	INDEXED & UNIQUE
48	OLE		
49	TEMPLATE_DEFINITIONSUNIQUEID {63}	TYPE:A	INDEXED
50	CUSTOMER_SALESCUSTOMER {1}	TYPE:A	INDEXED
51	INVENTORYSTOCK_NUM {9}	TYPE:A	INDEXED & UNIQUE
52	BUILD_CARDSEQ_NUM {1}	TYPE:L	INDEXED
53	LABELS		
54	ERP_HELPUNIQUEID {9}	TYPE:A	
55	INV_USAGE_MANYUNIQUEID {12}	TYPE:A	INDEXED
56	RESOURCE_CALENDARSLIP_NUMBER {11}	TYPE:L	INDEXED & UNIQUE
57	DELETELOGUNIQUEID {4}	TYPE:A	INDEXED & UNIQUE
58	DEPOSIT_ITEMSUNIQUEID {10}	TYPE:A	INDEXED & UNIQUE
59	CM_ITEMSUNIQUEID {10}	TYPE:A	INDEXED & UNIQUE

60	BOM_UNIQUEID {13}	TYPE:A	INDEXED
61	BOM_ITEMSUNIQUEID {20}	TYPE:A	INDEXED & UNIQUE
62	EMPLOYEECERTDEFUNIQUEID {20}	TYPE:A	
63	MISCDATAUNIQUEID {19}	TYPE:A	INDEXED
64	RM_REPORTSUNIQUEID {10}	TYPE:A	INDEXED
65	LOTCONTAINERSCONTAINERID {1}	TYPE:A	INDEXED & UNIQUE
66	MATERIALSUSEDUNIQUEID {17}	TYPE:A	INDEXED & UNIQUE
67	PO_ITEMSSEQNUM {15}	TYPE:L	INDEXED & UNIQUE
68	BO_APPROVERS2UNIQUEID {8}	TYPE:A	INDEXED
69	PARTDATAUNIQUEID {14}	TYPE:A	INDEXED & UNIQUE
70	WEB_MENU		
71	CURRENCYDATAUNIQUEID {7}	TYPE:A	
72	WEB_USER_SETTINGS		
73	DEBUGLOG		
74	BINNINGCONTROLUNIQUEID {8}	TYPE:A	INDEXED & UNIQUE
75	CODESTABLEUNIQUEID {6}	TYPE:A	INDEXED
76	AREADEFINITIONUNIQUEID {4}	TYPE:A	INDEXED & UNIQUE
77	CONSOLIDATEDSHIPPERUNIQUEID {2}	TYPE:A	INDEXED & UNIQUE
78	FORECASTDATAUNIQUEID {2}	TYPE:A	INDEXED
79	CUSTOMERRMASUNIQUEID {1}	TYPE:A	INDEXED & UNIQUE
80	RESERVATIONSUNIQUEID {1}	TYPE:A	INDEXED & UNIQUE
81	ACCTRANSACTIONUNIQUEID {3}	TYPE:A	INDEXED
82	MONTHLYGLBALANCEUNIQUEID {3}	TYPE:A	INDEXED & UNIQUE
83	ACCOUNTINGCONSTANTSUNIQUEID {3}	TYPE:A	INDEXED
84	MENUCONTROLUNIQUEID {1}	TYPE:A	INDEXED & UNIQUE
85	DEVICETABLEUNIQUEID {19}	TYPE:A	INDEXED & UNIQUE
86	CURRENCYQUOTESUNIQUEID {1}	TYPE:A	INDEXED & UNIQUE
87	SALESTAXTABLEUNIQUEID {1}	TYPE:A	INDEXED & UNIQUE
88	QCARTEAMUNIQUEID {4}	TYPE:A	INDEXED & UNIQUE
89	SLAVEDRIVENCHANGES		
90	TRIGGERMESSAGE		