Setting Up Carer Payments

last updated for the Spring 2013 (3.50) release

Technical Guide

Revision History

Version	Published on
Spring 2013 (3.50) - 1.0	26/02/2013

Doc Ref

Setting Up Carer Payments Technical Guide/Spring 2013 (3.50)/2013-02-26

Contacting the Service Desk

Please log a case via My Account

Providing Feedback on Documentation

We always welcome comments and feedback on the quality of our documentation including online help files and handbooks. If you have any comments, feedback or suggestions regarding the module help file, this handbook (PDF file) or any other aspect of our documentation, please email:

onepublications@capita.co.uk

Please ensure that you include the document name, version and aspect of documentation on which you are commenting.

.

Contents

01 | Carer Payments

Overview

The Carer Payments system consists of a few modules:

- the database (this is the One Database, used by v3 and v4 systems)
- the CCSCarerPaymentsProcessor application
- the MSMQ service

All these components must be installed/upgraded and configured properly for the Carer Payments system to run properly.

Database

The database must be upgraded using the standard update process, same as when a new release is deployed. The Carer Payments license must be applied to the database. After the system is licensed, the database will start generating entries in ExportQueue, whenever data important for Carer Payments is modified.

CCSCarerPaymentProcessor

The processor application has to be installed by running CCSCarerPaymentsSetup.msi file. It will install the exe application in a provided folder (default location is C:\Program Files\Capita Childrens Services\CCS Carer Payments Processor). The application file is called CCSCarerPaymentsProcessor.exe.

Before starting the application, a few things must be set up:

- the machine has to have “Message Queuing” installed (this is needed for sending the messages and may be installed using Windows Components Wizard)
- the configuration file has to be customised,
- the logger configuration file should be reviewed and modified to fulfil user’s needs

Configuration File

The configuration file by default is located in the Config subfolder of the folder where the application is installed. The file is called Config.ini. It has a few sections:

Database settings

The database credentials section format is same as in the main application server configuration file (so, if needed, the application server file may be used, after some modifications, as configuration for the carer payments processor).

Carer payments settings

This section is identified by “[CarerPayments]” header and has a few keys which have to be modified:

Key	Type	Description
CarerPaymentsQueue	String	Name of the main queue. In format expected by MSMQ: <Address>\[private$\]<QueueName> Where <Address> is the destination queue machine address and <QueueName> the destination queue name. The [private$\] part of the name is used when the queue is private.
AckQueue	String	Acknowledgment Queue name. Format is same as for the main queue above.
XSLTFilePath	String	Path to the XSLT file which is used for converting DB data into XML message acceptable by SoftBox. If not provided, the default XSLT file will be used.
UsersAllowed	String	List of users allowed to use the queues. This parameter is used only when the processor creates the queues or applies permissions to the queues.
MaxRetry	Number	Maximum number of times a message will be resent to MSMQ if the system does not get any confirmation about successful delivery of the message. When this limit is reached, the processor will log an error and will stop processing messages.

Logger Configuration File

The processor application can log messages in various ways and it allows the user to configure logging by modifying the logger.config file. For example, error messages may be sent via email and logged in the system Event Log, while all the messages are written to a log file. The default logger configuration file is as follows:

```
<LoggerConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <Logger xsi:type="CompositeLogger" logType="Information" strict="false">
          <Loggers>
               <!-- Log errors only to Windows Event log -->
               <Logger xsi:type="EventLogger" logType="Error" strict="true"
                applicationName="CCSCarerPaymentsProcessor" />

               <!-- Log all messages in the specified file -->
               <Logger xsi:type="FileLogger" logType="Information"
                strict="false">
                    <logfile>CCSCarerPaymentsProcessor.log</logfile>
               </Logger>
```

```
            <!-- To email error messages, uncomment the following section
             and fill in the details -->
            <!--
            <Logger xsi:type="MailLogger" logType="Error" strict="true"
             server="SMTP_ServerName_Or_IpAddress"
             from="username@server.co.uk">
                <Recipients>
                    <Recipient>recipient1@server.co.uk</Recipient>
                    <Recipient>recipient2@server.co.uk</Recipient>
                </Recipients>
            </Logger>
            -->
        </Loggers>
    </Logger>
</LoggerConfig>
```

The above configuration creates one logger which is a composite of two other loggers (the email logger is commented). One of them logs messages to a file (CCSCarerPaymentsProcessor.log) and the other one to the system Event Log (under the key named CCSCarerPaymentsProcessor). The following table explains the attributes of each.

Common attributes	
xsi:type	Type of a logger. At the moment three types are supported: EventLogger, FileLogger, MailLogger
logType	Specifies what messages will be logged. Each message has priority one of (from the highest): Error, Warning, Debug, Information. When the type is for example “Warning”, the logger will log warnings and errors (the specified priority plus all higher priority messages) by default.
strict	May be “true” or “false”. If strict, the logger will log only the messages of specified priority messages (logType), ignoring higher priority messages. If not strict, the logger will log the specified priority messages and all higher priority messages.
EventLogger attributes	
applicationName	Name of the application. It will be used in the system Event Log in the “Source” column of the “Application” section of the event log.
FileLogger attributes	
logfile	The file to log messages into. The path given is relative to the process working directory, so if just a file name is specified, the file will be created in the same folder where the processor runs.
MailLogger attributes	
server	Address of SMTP server which will be used for sending the messages.
from	Email address the emails will be sent from

Recipient	Email address of a recipient. Recipients are specified as a list of values (see the example file).

The email logger may be especially useful for sending error messages. Error messages are send by the processor only when an error is detected which stops the processor from continuing its work. Therefore it is rather important that errors are identified and addressed quickly. To create an email logger, the third logger section in the default file has to be uncommented and configuration details provided.

To add more loggers, one would have to copy the default section(s) and modify its attributes.

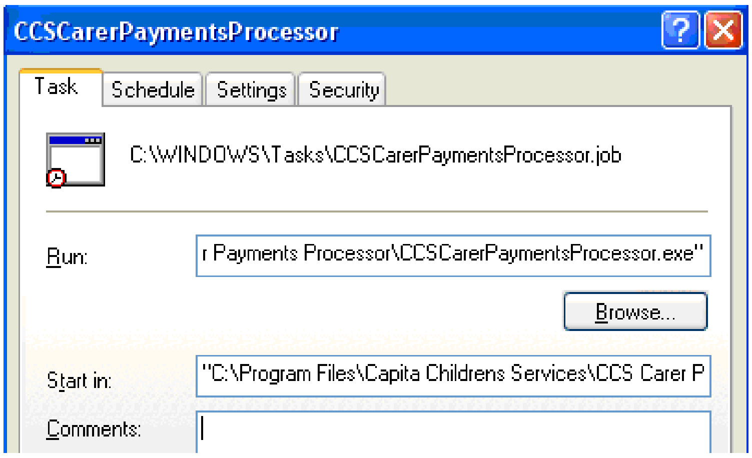
Command Line Arguments

The processor application accepts a few command line arguments and the following list describes them all.

Argument	Description
/HELP	Display the usage message
/APPLYRIGHTS [EXIT]	Set the queues user rights. It will take the user names from the configuration file (the UsersAllowed key). The optional EXIT parameter will stop the application from processing the queues, only the rights will be applied. When the queues are not present on the MSMQ server(s), the process will create the queues too.
/INITIAL	Start the initial export of placement records
/FORCE	Ignore the MaxRetry parameter in the configuration file. Force the application to process the queues and messages even when the maximum number of a message being resent is exceeded. This option is useful when the processor stops due to the MaxRetry problem and the whole system must be started manually after sorting out any issues causing the problem.
/STARTDATE <date>	Send all placement records which were modified after the given date, or are active on the date or later, or are waiting in the export queue. The date is in ShortDate system format (displayed when the /HELP argument is used. For more information on the symbols used, please see *DateTimeFormatInfo Class* on the Microsoft website (http://msdn.microsoft.com/en-us/library/system.globalization.datetimeformatinfo.aspx). This option is useful after the carer payments system (triggers especially) is down for some reason (e.g. missing license) and the processor has to send “missing” messages.

Scheduling the process

Normally, the processor application does not need any arguments, after it is configured properly and the initial export is done. To run the process e.g. every 15 minutes, one would have to schedule the application running every 15 minutes, just calling the exe program, with no other arguments. The windows scheduler (Scheduled Tasks) can be used to do the job (see screen print below):



C:\WINDOWS\Tasks\CCSCarerPaymentsProcessor.job

Run: r Payments Processor\CCSCarerPaymentsProcessor.exe"

Start in: "C:\Program Files\Capita Childrens Services\CCS Carer P

Comments:

MS Message Queue

The MSMQ service must be installed on the machine which hosts the queues and on the one where the CCSCarerPaymentsProcessor runs on (it may be the same server, but does not have to). During the installation, the “Active Directory Integration” mode should be selected (not the workgroup mode).

Once the service is installed on appropriate machines, the processor application may be used to create the queues. There are two of them: main queue and acknowledgment queue. The main queue should be created on the MSMQ server, the acknowledgment queue on the server where the processor runs.

The processor application may be used to create the queues and set up their permissions (/APPLYRIGHTS option). To get more information about it, look into the Quick Setup Procedure section of this document.

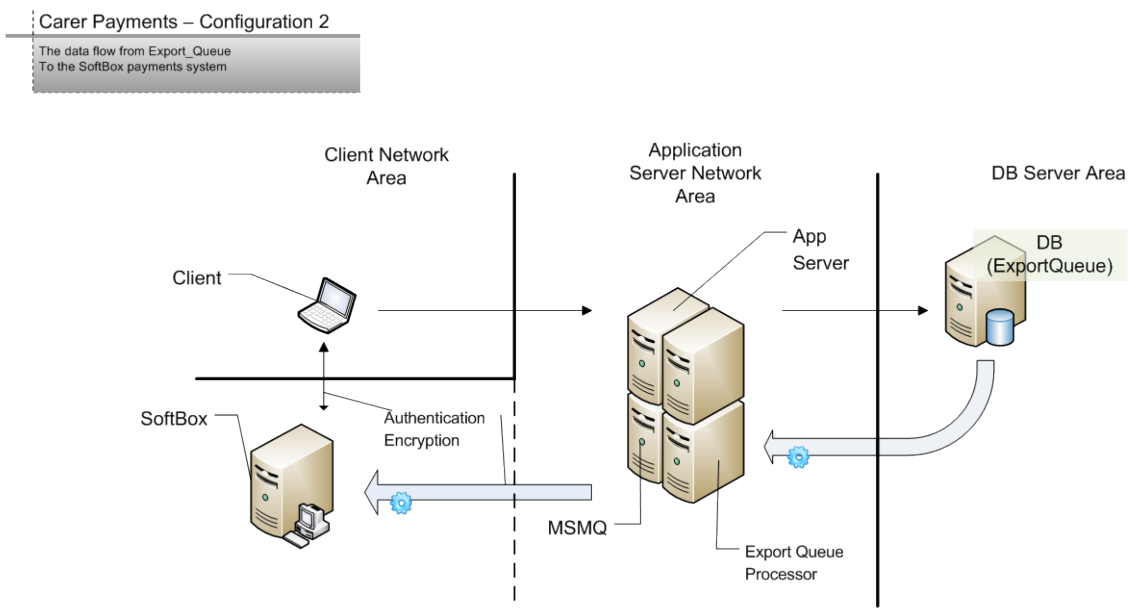
Softbox will request to read the payments processor messages from msmq. To this end, appropriate permissions should be granted to the softbox user.

Deployment of Carer Payment Components

The Carer Payment module may be deployed in a few parts, each component in a different area of the network:

Area	Access Description	Components
DB Server Area	No component, except the application server, has access to this area.	DB server (triggers and export queue), Export Queue Processor (which connects to the DB)
Application Server Area	Client applications have access to this area.	Application server, MSMQ and SoftBox server. The SoftBox server behaves as a client of the application server area, but is a server to the users. MSMQ may be placed in a separate subarea (Server) from the application server too, if needed.
Client Area	Everyone has access	ONE clients and clients of the SoftBox server.

There is another scenario possible, if the customer does not want to deploy all the components on different machines. The ExportQueue Processor and MSMQ may be placed on the application server:


Carer Payments – Configuration 2
The data flow from Export_Queue
To the SoftBox payments system
Client Network Area
Application Server Network Area
DB Server Area
Client
App Server
DB (ExportQueue)
SoftBox
Authentication Encryption
MSMQ
Export Queue Processor

Quick Setup Procedure

1. Read the section in this document titled ‘Deployment of Carer Payment Components’, so that you are familiar with the different architecture layers utilised by Carer Payments.
2. Backup the database.
3. Upgrade the database, applying appropriate build, containing the CarerPayments module changes.
4. Add the CarerPayments license to the database.
5. Install MSMQ service on all machines where MSMQ queues are processed (the MSMQ service machine, the payments processor machine)
6. Install the CCSCarerPaymentsSetup.msi package on the processor machine.
7. Open the processor configuration file (Config/Config.ini in the folder where the processor was installed)
8. In the [CarerPayments] section:
   a. Edit the CarerPaymentsQueue key, providing the main queue name. It is the queue SoftBox will read data from. It should be located on the MSMQ server, e.g. CRP_MSMQ\CarerPaymentsQueue
   b. Edit the AckQueue key, providing the name of the acknowledgment queue, used by the processor to get confirmations about successful delivery of messages. It should be located on the processor machine (where the config file is), e.g. .\ CarerPaymentsAckQueue
   c. Add users to the UsersAllowed key, for example: UsersAllowed = Domain\processor; Domain\softbox
   d. Add value to the MaxRetry key, specifying how many times the system should try resending a message before failing and stopping. The default value is 5.
9. Save the file, close it and run the processor with the “ApplyRights” option, adding the EXIT parameter. It will create the queues and apply permissions:
```
> CCSCarerPaymentsProcessor.exe /APPLYRIGHTS EXIT
```
10. Check the log file for any errors.
11. Run the initial export:
```
> CCSCarerPaymentsProcessor.exe /INITIAL
```
12. Check the log file for any errors. Check the main queue, whether the messages are waiting on the main queue.
13. Run the processor without any arguments, checking that it runs properly in the “normal” mode:

```
> CCSCarerPaymentsProcessor.exe
```

14. Set up the windows scheduler, to run the processor periodically. When running the process in the “normal” mode, no arguments are needed, just the exe application started.