DDCM API
User Guide

V10.0

CONTENTS

Introduction

The DDCM API allows a way for companies to interact directly with DDCM to create mandates, process collections and receive reports.

In order to use the API users will first need to be approved and enabled to RSM. If you wish to use this feature please contact RSM support.

API Interface

Sending A Request

An XML request needs to be sent as part of a secure HTTP POST, all xml data must be put into a single parameter with the name ‘xml’.

XML requests will have the following format:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "apiDtd/apiCommand.dtd">
<ddcm>
   <request>
      <username>rsm</username>
      <command>apiCommand</command>
      <commandRequestParameters>...</commandRequestParameters>
      <timestamp>0000000000</timestamp>
   </request>
   <signature />
</ddcm>
```

A typical request header would look like this:

```
POST https://rsm5.rsmsecure.com/ddcm/ddcmApi.php HTTP/1.1

Host: rsm5.rsmsecure.com

Content-Length: 617

Content-Type: application/x-www-form-urlencoded

xml=%3C%3Fxml+version%3D%221.0%22+encoding%3D%22ISO-8859-
1%22%3F%3E%3C%21DOCTYPE+ddcm+SYSTEM+%22apiDtd%2FheartbeatRequest.dtd%22%3E%3C
ddcm%3E%0D%0A%09%3Crequest%3E%0D%0A%09%09%3Ccommand%3Eheartbeat%3C%2Fcommand%
3E%0D%0A%09%09%3Cusername%3Eadmin%3C%2Fusername%3E%0D%0A%09%09%3Ctimestamp%3E
1508516449%3C%2Ftimestamp%3E%0D%0A%09%3C%2Frequest%3E%0D%0A%09%3Csignature%3E
f58a713a860b1abfec921cbb6122e37025a8d7881937c60b1493f4a26b8cac3e%3C%2Fsignatu
re%3E%0D%0A%3C%2Fddcm%3E
```

API requests should be sent to the following URLs:

Live Server

https://rsm5.rsmsecure.com/ddcm/ddcmApi.php

Demo Server

https://rsm5.rsmsecure.com/demo/ddcm/ddcmApi.php

The demo system also has a test page which can be used to quickly generate and test XML requests on the demo DDCM system: *https://rsm5.rsmsecure.com/demo/ddcm/testDdcmApi.html*

Receiving A Response

The response format will always be an XML v1.0 ISO-8859-1 encoded document with a root node called <ddcm>, for example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "https://rsm5.rsmsecure.com /ddcm/apiDtd/response.dtd">
<ddcm>
   <response>
      <status />
      <commandResponseParameters />
   </response>
   <signature />
</ddcm>
```

Generating The Signature

The <signature> element is used to verify the integrity of the request; it is generated by SHA-256 hashing the request with the users set API password appended. **Important: There must be no spaces, line breaks or tabs between elements. Also, there must be no spaces before the closing tags of empty elements e.g. <phone/> not <phone />.**

For example here is how a signature could be generated for a heart beat request.

Step One

Generate the XML for the request:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "apiDtd/heartbeatRequest.dtd">
<ddcm>
   <request>
      <command>heartbeat</command>
      <username>admin</username>
      <timestamp>1508513754</timestamp>
   </request>
</ddcm>
```

Step Two

Dump the request element to a string, ensure that there is no whitespace or carriage returns between elements:

```
<request><command>heartbeat</command><username>admin</username><timestamp>150
8513754</timestamp></request>
```

Step Three

Append the API password to the end of the string:

```
<request><command>heartbeat</command><username>admin</username><timestamp>150
8513754</timestamp></request>1234567890
```

Step Four

Hash the string using SHA-256, the returned value is the signature:

```
3274f2273abf50e8b31a9fe4aca6874546fe9f5d57d37bcc79165c11e6a24da0
```

Step five

Add the generated signature to the ddcm element:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "apiDtd/heartbeatRequest.dtd">
<ddcm>
   <request>
      <command>heartbeat</command>
      <username>admin</username>
      <timestamp>1508513754</timestamp>
   </request>

<signature>3274f2273abf50e8b31a9fe4aca6874546fe9f5d57d37bcc79165c11e6a24da0</
signature>
</ddcm>
```

If a request does not contain a signature or the signature given do not match the signature generated by the API then the entire request will fail.

The response will also contain a <signature> element which can be used to verify its integrity. The response signature is generated in the same way as the request signature.

XML DTD

All requests are validated against a DTD to confirm that they are well formed before attempting to process the request. If a DTD is not included the request will fail with the error 'request does not match given DTD'

Below is a list of all the DTD's used for each API command and response.
The DTD URL is what should be used if you wish to download or validate the XML before sending the request to the API. The 'DTD request location in XML' should be the path that's used in the actual XML request sent to the API. All Responses will contain the DTD URL

Command Name	DTD URL	DTD Request location in XML
Heartbeat request	*https://rsm5.rsmsecure.com/ddcm/apiDtd/heartbeatRequest.dtd*	apiDtd/heartbeatRequest.dtd
Heartbeat response	*https://rsm5.rsmsecure.com/ddcm/apiDtd/heartbeatResponse.dtd*	
setMandates request	*https://rsm5.rsmsecure.com/ddcm/apiDtd/setMandatesRequest.dtd*	apiDtd/setMandatesRequest.dtd
setMandates response	*https://rsm5.rsmsecure.com/ddcm/apiDtd/setMandatesResponse.dtd*	
setCollections request	*https://rsm5.rsmsecure.com/ddcm/apiDtd/setCollectionsRequest.dtd*	apiDtd/setCollectionsRequest.dtd
setCollections response	*https://rsm5.rsmsecure.com/ddcm/apiDtd/setCollectionsResponse.dtd*	
mandateReport request	*https://rsm5.rsmsecure.com/ddcm/apiDtd/mandateReportRequest.dtd*	apiDtd/mandateReportRequest.dtd
mandateReport response	*https://rsm5.rsmsecure.com/ddcm/apiDtd/mandateReportResponse.dtd*	
collectionReport request	*https://rsm5.rsmsecure.com/ddcm/apiDtd/collectionReportRequest.dtd*	apiDtd/collectionReportRequest.dtd
collectionReport response	*https://rsm5.rsmsecure.com/ddcm/apiDtd/collectionReportResponse.dtd*	
mandateBatch Report request	*https://rsm5.rsmsecure.com/ddcm/apiDtd/mandateBatchReportRequest.dtd*	apiDtd/mandateBatchReportRequest.dtd
mandateBatch Report	*https://rsm5.rsmsecure.com/ddcm/apiDtd/mandateBatc*	

response	*hReportResponse.dtd*	
collectionBatch Report request	*https://rsm5.rsmsecure.com/ddcm/apiDtd/collectionBatchReportRequest.dtd*	apiDtd/collectionBatchReportRequest.dtd
collectionBatch Report response	*https://rsm5.rsmsecure.com/ddcm/apiDtd/collectionBatchReportResponse.dtd*	
letterBatchReport request	*https://rsm5.rsmsecure.com/ddcm/apiDtd/letterBatchReportRequest.dtd*	apiDtd/letterBatchReportRequest.dtd
letterBatchReport response	*https://rsm5.rsmsecure.com/ddcm/apiDtd/letterBatchReportResponse.dtd*	
collectionStatement request	*https://rsm5.rsmsecure.com/ddcm/apiDtd/collectionStatementRequest.dtd*	apiDtd/collectionStatementRequest.dtd
collectionStatement response	*https://rsm5.rsmsecure.com/ddcm/apiDtd/collectionStatementResponse.dtd*	
monthlyStatement request	*https://rsm5.rsmsecure.com/ddcm/apiDtd/monthlyStatementRequest.dtd*	apiDtd/monthlyStatementRequest.dtd
monthlyStatement response	*https://rsm5.rsmsecure.com/ddcm/apiDtd/monthlyStatementResponse.dtd*	
responseStatus Check request	*https://rsm5.rsmsecure.com/ddcm/apiDtd/responseStatusCheckRequest.dtd*	apiDtd/responseStatusCheckRequest.dtd

Commands

The API supports the commands listed below. The subsections following give more detail about each command.

Command Name	Description
heartbeat	Used to check system availability.
setMandates	Add a new mandate or edit/cancel an existing mandate.
setCollections	Add a new collection for an existing mandate or edit/delete existing collection.
mandateReport	Retrieves a report listing existing mandates based on the filter options sent as part of the request.
collectionReport	Retrieves a report listing existing collections based on the filter options sent as part of the request.
mandateBatchReport	Retrieves a report listing existing mandate batches based on the filter options sent as part of the request.
collectionBatchReport	Retrieves a report listing existing collection batches based on the filter options sent as part of the request.
letterBatchReport	Retrieves a report listing existing letter batches based on the filter options sent as part of the request.
collectionStatement	Retrieves collection statement based on the collection or statement date.
monthlyStatement	Retrieves monthly statement based on statement month.
responseStatusCheck	Retrieves the response returned by the command previously executed. Supports ‘setMandates’ and ‘setCollections’.

Note: Users must have appropriate permission to use each of these commands. The permissions are set by RSM support staff.

Heartbeat

Used to check if the system is online and the user is correctly set up to use the API.

Request

Element Name	Details	Optional	Validation	Character Range	Typical Value
<command>	The name of the API command.	N			heartbeat
<username>	The username required to identify the user and vendor.	N	Alphanumeric	4-40	Testuser
<timestamp>	Unix timestamp of when the request was made.	N	Numeric		1377791226

Response

Element Name	Description
<status>	Holds the status of the heartbeat request, if the system is available and the vendor is set up properly returns SUCCESS otherwise FAIL is returned.
<errors>	All errors will be contained in the errors element. This element is empty if the request was successful.
└ **<error>**	An element for each error returned.
└ **<code>**	Numeric error code.
└ **<detail>**	Human readable error message that describes what the exact error is.

Example

The following is an example of a typical heartbeat command:

```
XML REQUEST:

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "apiDtd/heartbeatRequest.dtd">
<ddcm>
   <request>
      <command>heartbeat</command>
      <username>testUser</username>
      <timestamp>0000000000</timestamp>
   </request>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

The following response would be returned:

```
XML RESPONSE:

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM
"https://rsm5.rsmsecure.com/ddcm/apiDtd/heartbeatRequest.dtd">
<ddcm>
   <response>
      <status>FAIL</status>
      <errors>
         <error>
            <code>1001</code>
            <detail>Invalid Request</detail>
         </error>
      </errors>
   </response>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

Set Mandates

Used add a new mandate to the DDCM and also used to amend or cancel an existing mandate.

Request

Element Name	Details	Optional	Validation	Character Range	Typical Value
<command>	The name of the API command.	N			setMandates
<username>	The username required to identify the user and vendor.	N	Alphanumeric	4-40	Testuser
<mandates>					
└ **<mandate>**					
└ **<tradingName>**	Trading name.	Y	Alphanumeric	0-50	testName
└ **<contactName>**	Contact name.	Y	Alphanumeric	0-50	lastName
└ **<address1>**	Address line 1.	N – Required if address checking is enabled.	Alphanumeric	0-30	Housename
└ **<address2>**	Address line 2.	Y	Alphanumeric	0-30	streetName
└ **<address3>**	Address line 3.	Y	Alphanumeric	0-30	addressLine
└ **<town>**	Town/City.	N – Required if address checking is enabled.	Alphanumeric	0-30	xyzTown
└ **<postcode>**	Postcode.	N – Required if address checking is enabled.	Alphanumeric	0-12	XY12 3ZT
└ **<phone>**	Telephone number.	Y	Numeric and the following characters ( ) + -	0-20	0123456789

└ **<email>**	Email address.	N – Required if confirmation emails are mandatory.	All characters allowed, must be formatted like a valid email address.	0-50	test@rsm2000.co.uk
└ **<clientRef>**	Client’s reference number.	N	Alphanumeric	0-50	testRSM
└ **<accountName>**	Account holders name.	N	Alphanumeric	0-50	testAccount
└ **<accountNumber>**	Account number.	N	Numeric	8	12345678
└ **<sortCode>**	Sort Code.	N	Numeric	6	123456
└ **<action>**	Determines what action performed on the mandate: N=New A=Amend D=Cancel	N	N, A or D	1	N
└ **<ddRefNo>**	BACS Direct Debit reference number.	N – Required only while adding a new mandate and if DDRefNo number are not automatically generated.	Alphanumeric	6-18	testDDRef123
└ **<amount>**	Amount for repeating collections.	Y – Only required for repeating mandates. If sent both frequency & startDate are required.	Floating point to either 2 or 0 D.P.	1-9	120.05 15

└ **<frequency>**	Frequency of which a repeating mandate would make a collection.	Y – Only required for repeating mandates. If sent both amount & startDate are required.	Numeric	1,3,6 or 12	1
└ **<startDate>**	Date on which a repeating mandate should make its first collection.	Y – Only required for repeating mandates. If sent both amount & frequency are required.	Date DD/MM/YY or DD/MM/YYYY Should be at least 5 working days in the future and not later than 13 months.	1-10	15/08/2013 15/08/13
└ **<mandateType>**	Mandate type if enabled for vendor.	N – Required only if configurable mandate types are enabled.	Alphanumeric	2	E L
└ **<shortId>**	Short id of the sub vendor if a super vendor is adding mandates on behalf of their sub vendors.	N – Required only for super vendors.	Alphanumeric	1-10	125 testShort
└ **<endDate>**	Date after which no more collections are taken by a repeating mandate.	Y – If sent amount, frequency, & startDate are all required. Only applicable for repeating mandates.	Date DD/MM/YY DD/MM/YYYY Should not be later than the start date.	1-10	15/08/2025
└ **<paymentRef>**	Payment	Y	Alphanumeric	0-30	Tstref1

	reference.			
<timestamp>	Unix timestamp of when the request was made.	N	Numeric	1377791226

RESPONSE

Element Name	Description
<status>	Holds the status of the 'setMandates' request. If the mandate action was performed successfully the API returns SUCCESS otherwise FAIL is returned.
<summary>	
└ **<totalSubmitted>**	Total number of mandates that were submitted as part of the request.
└ **<totalSuccessful>**	Total number of mandates that were successfully updated by the system.
└ **<totalFailed>**	Total number of mandates that had errors while updating.
<mandates>	
└ **<mandate>**	An element for each mandate submitted as part of the request.
└ **<status>**	SUCCESS if the update was successful otherwise FAIL is returned.
└ **<ddRefNo>**	The DDRefNo of the mandate submitted.
└ **<clientRef>**	The ClientRef of the mandate submitted.
└ **<errors>**	
└ **<error>**	
└ **<code>**	Numeric error code.
└ **<detail>**	Human readable error message that describes what the exact error is.

Example

The following is an example of a typical ‘setMandates’ command:

XML REQUEST:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "apiDtd/setMandatesRequest.dtd">
<ddcm>
   <request>
      <command>setMandates</command>
      <username>xyzergyd</username>
      <mandates>
         <mandate>
            <tradingName>Trading Name</tradingName>
            <contactName>Last name</contactName>
            <address1>House Number</address1>
            <address2>Street</address2>
            <address3>Address Line</address3>
            <town>Bedford</town>
            <postcode>TE1 3ST</postcode>
            <phone>0123456789</phone>
            <email>test@rsm2000.co.uk</email>
            <clientRef>test1234</clientRef>
            <accountName>Account Name</accountName>
            <accountNumber>12345678</accountNumber>
            <sortCode>123456</sortCode>
            <action>N</action>
            <ddRefNo>test1234</ddRefNo>
            <amount>101</amount>
            <frequency>1</frequency>
            <startDate>15/08/2013</startDate>
            <mandateType>L</mandateType>
            <shortId>SID</shortId>
            <endDate>15/08/2023</endDate>
            <paymentRef>TestRef</paymentRef>
         </mandate>
      </mandates>
      <timestamp>0000000000</timestamp>
   </request>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

The following response would be returned:

XML RESPONSE:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM
"https://rsm5.rsmsecure.com/ddcm/apiDtd/setMandatesResponse.dtd">
<ddcm>
   <response>
      <status>FAIL</status>
      <summary>
         <totalSubmitted>1</totalSubmitted>
         <totalSuccessful>0</totalSuccessful>
         <totalFailed>1</totalFailed>
      </summary>
      <mandates>
         <mandate>
            <status>FAIL</status>
            <ddRefNo>test1234</ddRefNo>
            <clientRef>test1234</clientRef>
            <errors>
               <error>
                  <code>1001</code>
                  <detail>Invalid Request</detail>
               </error>
            </errors>
         </mandate>
      </mandates>
   </response>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

Set Collections

Used to add, edit or delete the collections on the DDCM system.

Request

Element Name	Details	Optional	Validation	Character Range	Typical Value
<command>	The name of the API command.	N			setCollections
<username>	The username required to identify the user and vendor.	N	Alphanumeric	4-40	Testuser
<collections>					
└ **<collection>**					
└ **<sun>**	Service user number.	N	Numeric	6	123456
└ **<clientRef>**	Client's reference number.	N	Alphanumeric	0-50	testRSM
└ **<amount>**	Collection amount.	N	Floating point to either 2 or 0 D.P.	1-9	120.05 25
└ **<date>**	Collection date.	N	Date in the format DD/MM/YY or DD/MM/YYYY Should be at least 3 working days in the future and not later than 13 months.	1-10	15/08/2013 15/08/13
└ **<ddRefNo>**	BACS Direct Debit reference number.	Y	Alphanumeric	6-18	testDDRef123

└ **<action>**	Field to determine what action is to be performed against the collection. N=New A=Amend D=Delete	Y – Default value of N	N, A or D	1	N
└ **<existingAmount>**	The current collection amount used to lookup the collection when editing.	Y – required if action is A	Floating point to either 2 or 0 D.P.	1-9	120.05 25
└ **<existingDate>**	The current collection date used to lookup the collection while editing.	Y – required if action is A	Date in the format DD/MM/YY or DD/MM/YYYY Should be at least 3 working days in the future and not later than 13 months.	1-10	15/08/2013 15/08/13
<timestamp>	Unix timestamp when the request was made.	N	Numeric		1377791226

Response

Element Name	Description
<status>	Status of the 'setCollections' request. If the collection action was performed successfully the API returns SUCCESS otherwise FAIL is returned.
<summary>	
└ **<totalSubmitted>**	Total number of collections that were submitted as part of the request.
└ **<totalSuccessful>**	Total number of collections that were successfully updated by the system.
└ **<totalFailed>**	Total number of collections that had errors while updating.
<collections>	
└ **<collection>**	
└ **<status>**	SUCCESS if the update was successful , FAIL if it failed.
└ **<ddRefNo>**	The DDRefNo of the collection submitted.
└ **<clientRef>**	The ClientRef of the collection submitted.
└ **<errors>**	
└ **<error>**	
└ **<code>**	Numeric error code.
└ **<detail>**	Human readable error message that describes what the exact error is.

EXAMPLE

The following is an example of a typical ‘setCollections’ command:

```
XML REQUEST:

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "apiDtd/setCollectionsRequest.dtd">
<ddcm>
   <request>
      <command>setCollections</command>
      <username>xyzergyd</username>
      <collections>
         <collection>
            <sun>123456</sun>
            <clientRef>test1234</clientRef>
            <amount>101</amount>
            <date>15/08/2013</date>
            <ddRefNo>test1234</ddRefNo>
            <action>N</action>
            <existingAmount>15</existingAmount>
            <existingDate>14/08/2013</existingDate>
         </collection>
      </collections>
      <timestamp>0000000000</timestamp>
   </request>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

The following response would be returned:

XML RESPONSE:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM
"https://rsm5.rsmsecure.com/ddcm/apiDtd/setCollectionsResponse.dtd">
<ddcm>
   <response>
      <status>FAIL</status>
      <summary>
         <totalSubmitted>1</totalSubmitted>
         <totalSuccessful>0</totalSuccessful>
         <totalFailed>1</totalFailed>
      </summary>
      <collections>
         <collection>
            <status>FAIL</status>
            <ddRefNo>test1234</ddRefNo>
            <clientRef>test1234</clientRef>
            <errors>
               <error>
                  <code>1001</code>
                  <detail>Invalid Request</detail>
               </error>
            </errors>
         </collection>
      </collections>
   </response>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

Mandate Report

Request

Element Name	Details	Optional	Validation	Character Range	Typical Value
<username>	The username required to identify the user and vendor.	N	Alphanumeric	4-40	Testuser
<command>	The name of the API command.	N			mandateReport
<filters >					
└ **<status>**	Used to filter mandates by their current status. If only the status filter is used only mandates modified or created on the current date will be returned.	Y	Alpha	LIVE NEW PENDING FAILED AMEND RESUBMIT CANCEL CANCELLED DELETED	LIVE
└ **<startDate>**	Starting date of range you wish to filter by. This will include all mandates modified or created in this range.	Y	Valid date in the format DD/MM/YYYY.		25/10/2013
└ **<endDate >**	Ending date of range you wish to filter by. Required if the start date has	Y	Valid date in the format DD/MM/YYYY.		29/11/2013

	been populated.				
└ **<ddRefNo >**	BACS Direct Debit Reference Number. Used to filter the mandate report to any mandates matching the given DD reference number. You can also perform lookups using partial DD references.	Y	Alphanumeric	6-18	testDDRef123
└ **<clientRef >**	Client’s reference number. Used to filter the mandate report to any mandates matching the given client reference. You can also perform lookups using partial client references.	Y	Alphanumeric	0-50	testRSM
└ **<postcode >**	Postcode. Used to filter the mandate report to any mandates matching the given postcode. You can also perform lookups using partial postcodes	Y	Alphanumeric	0-12	XY12 3ZT
└ **<name >**	Account holder name. Used to filter the	Y	Alphanumeric	0-50	testName

	mandate report to any mandates matching the given account holder name. You can also perform lookups using partial names.				
└ **<reason>**	Fail reason Used to filter the mandate report to any mandates matching the given fail reason. You can also perform lookups using partial reasons.	Y	Alphanumeric	0-50	0 1:
└ **<mandateType>**	Mandate type Used to filter the mandate report to any mandates matching the given mandate type. You can also perform lookups using partial types. This filter is only applicable for vendor's who have mandate types enabled.	Y	Alphanumeric	2	E L
└ **<shortId>**	Short id. Used to filter the mandate report to any mandates matching the given short id. You can also	Y	Alphanumeric	1-10	12 42

	perform lookups using partial short id’s. This filter is only applicable for super vendors.				
└ **<paymentRef>**	Payment reference. Used to filter the mandate report to any mandates matching the given payment ref	Y	Alphanumeric	1-30	Testref1
└ **<sortType>**	Field on which the report is to be sorted.	Y – default is DDRefNo	Alpha	1-10	name clientRef
└ **<sortOrder>**	Order in which the report is to be sorted. Ignored if sortType is not provided.	Y – default is ASC	Alpha	ASC DESC	ASC DESC
└ **<noOfRows>**	The number of mandates returned by the report. Value must be a multiple of 5 otherwise it will default to 20.	Y – Default is blank, which would retrieve all the rows returned by the search.	Numeric	0-4	20 100
<timestamp>	Unix timestamp when the request was made.	N	Numeric		1377791226

Response

Element Name	Description
<status>	Holds the status of the 'mandateReport' request. If the collection action was performed successfully the API returns SUCCESS else returns FAIL.
<errors>	All the errors would be shown in the errors element. This element is empty if the request was successful.
└ **<error>**	
└ **<code>**	Numeric error code
└ **<detail>**	Human readable error message that describes what the exact error is.
<noOfRecords>	Number of mandate records returned as part of the response. If no records are returned the status is SUCCESS and noOfRecords is 0.
<mandates>	
└ **<mandate>**	
└ **<ddRefNo>**	BACS Direct Debit reference number.
└ **<clientRef>**	Client reference.
└ **<name>**	Account holder's name.
└ **<sortcode>**	Sort code.
└ **<account>**	Account number.
└ **<postcode>**	Postcode.
└ **<amount>**	Amount .
└ **<startDate>**	Start date in YYYY-MM-DD format
└ **<frequency>**	Pay frequency.
└ **<endDate>**	End date in YYYY-MM-DD format.
└ **<updated>**	Date on which the mandate was last updated in YYYY-MM-DD format.
└ **<status>**	The current status of the mandate.
└ **<failReason>**	Mandate failure reason.
└ **<shortId>**	Short id of the vendor to which the mandate belongs to. Only returned for

	super vendors.
└ **<mandateType>**	Mandate type. Only returned if the vendor has mandate types enabled.
└ **<fullClientRef>**	Full client reference. Only returned for super vendors.
└ **<vendor>**	Vendor’s short name. Only returned for super vendors.

The above table describes the standard DDCM response for ‘mandateReport ‘request. If the vendor chooses to customise the mandate report by logging into the DDCM system they may get different fields in a different order based on the changes.

Example

The following is an example of a typical ‘mandateReport’ command:

XML REQUEST:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "apiDtd/mandateReportRequest.dtd">
<ddcm>
   <request>
      <command>mandateReport</command>
      <username>xyzergyd</username>
      <filters>
         <status>ALL</status>
         <startDate>01/01/2013</startDate>
         <endDate>01/08/2013</endDate>
         <ddRefNo>test1234</ddRefNo>
         <clientRef />
         <postcode />
         <name />
         <reason />
         <mandateType />
         <shortId />
         <paymentRef />
         <sortType>updated</sortType>
         <sortOrder>ASC</sortOrder>
         <noOfRows>20</noOfRows>
      </filters>
      <timestamp>0000000000</timestamp>
   </request>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

The following response would be returned:

XML RESPONSE:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM
"https://rsm5.rsmsecure.com/ddcm/apiDtd/mandateReportResponse.dtd">
<ddcm>
   <response>
      <status>SUCCESS</status>
      <errors />
      <noOfRecords>1</noOfRecords>
      <mandates>
         <number>1</number>
         <mandate>
            <ddRefNo>test1234</ddRefNo>
            <clientRef>test1234</clientRef>
            <name>Account Name</name>
            <sortcode>123456</sortcode>
            <account>12345678</account>
            <postcode>TE1 3ST</postcode>
            <amount>101</amount>
            <startDate>2013-08-15</startDate>
            <frequency>1</frequency>
            <endDate>2023-08-15</endDate>
            <updated>2013-06-01</updated>
            <status>LIVE</status>
            <failReason />
            <shortId>SID</shortId>
            <mandateType>L</mandateType>
            <vendor>testVendor</vendor>
         </mandate>
      </mandates>
   </response>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

Collection Report

Request

Element Name	Details	Optional	Validation	Character Range	Typical Value
<username>	The username required to identify the user and vendor.	N	Alphanumeric	4-40	Testuser
<command>	The name of the API command.	N			collectionReport
<filters >					
└ **<status>**	Used to filter collection records by their current status. If only the status filter is used only collections made on the current date will be returned.	Y	Alpha	SUCCESS FAILED NEW UNPAID PAID First collection Normal collection	NEW
└ **<startDate>**	Starting date of range you wish to filter by.	Y	Valid date in the format DD/MM/YYYY		25/10/2013
└ **<endDate >**	Ending date of range you wish to filter by. Required if the start date has been populated.	Y	Valid date in the format DD/MM/YYYY		29/11/2013
└ **<ddRefNo >**	BACS Direct Debit Reference number. Used to filter the collection report to any collections	Y	Alphanumeric	6-18	testDDRef123

	matching the given DD reference number. You can also perform lookups using partial DD references.				
└ **<clientRef >**	Client’s reference number. Used to filter the collection report to any collections matching the given client reference. You can also perform lookups using partial client references.	Y	Alphanumeric	0-50	testRSM
└ **<name >**	Account holder name. Used to filter the collection report to any collections matching the given account holder name. You can also perform lookups using partial names.	Y	Alphanumeric	0-50	testName
└ **<sortType>**	Field on which the report is to be sorted.	Y	Alpha	1-10	name clientRef
└ **<sortOrder>**	Order in which the report is to be sorted. Ignored if sortType is not provided.	Y – default is ASC	Alpha	ASC DESC	ASC DESC
└ **<noOfRows>**	The max number of collections to be	Y	Numeric	0-4	20

	returned. If this is not set all matches are returned. Value must be a multiple of 5 otherwise it will default to 20.			100
<timestamp>	Unix timestamp when the request was made.	N	Numeric	1377791226

Response

Element Name	Description
<status>	Holds the status of the 'collectionReport' request. If the collection action was performed successfully the API returns SUCCESS otherwise FAIL is returned.
<errors>	All the errors would be shown in the errors element. This element is empty if the request was successful.
└ **<error>**	
└ **<code>**	Numeric error code.
└ **<detail>**	Human readable error message that describes what the exact error is.
<noOfRecords>	Number of collection records returned as part of the response. If no records are returned the status is SUCCESS and noOfRecords is 0.
<collections>	
└ **<collection>**	
└ **<dateDue>**	The collection date in YYYY-MM-DD format.
└ **<ddRefNo>**	BACS Direct Debit reference number.
└ **<clientRef>**	Client reference.
└ **<name>**	Account holder's name.
└ **<sortcode>**	Sort code.
└ **<account>**	Account number.
└ **<amount>**	Amount.
└ **<repeat>**	Y if the collection is repeating else N.
└ **<status>**	The current status of the collection.
└ **<failReason>**	Collection failure reason.
└ **<payStatus>**	Pay status of the collection.
└ **<paidAmount>**	Paid amount of the collection.
└ **<collectionType>**	Collection type based on if it is first, regular, last collection or an indemnity (for repeating collections).

└ **<shortId>**	Short id of the vendor to which the collection belongs to. Only returned for super vendors.
└ **<vendor>**	Vendor’s short name. Only returned for super vendors.
└ **<mandatePayRef>**	The payment reference of the mandate at the time the collection was created.

The above table describes the standard DDCM response for ‘collectionReport’ request. If the vendor chooses to customise the collection report by logging into the DDCM system they may get different fields in a different order based on the changes.

EXAMPLE

The following is an example of a typical ‘collectionReport’ command:

```
XML REQUEST:

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "apiDtd/collectionReportRequest.dtd">
<ddcm>
   <request>
      <command>collectionReport</command>
      <username>xyzergyd</username>
      <filters>
         <status>ALL</status>
         <startDate>01/01/2013</startDate>
         <endDate>01/08/2013</endDate>
         <ddRefNo>test1234</ddRefNo>
         <clientRef />
         <name />
         <sortType>date</sortType>
         <sortOrder>ASC</sortOrder>
         <noOfRows>20</noOfRows>
      </filters>
      <timestamp>0000000000</timestamp>
   </request>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

The following response would be returned:

XML RESPONSE:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM
"https://rsm5.rsmsecure.com/ddcm/apiDtd/collectionReportResponse.dtd">
<ddcm>
   <response>
      <status>SUCCESS</status>
      <errors />
      <noOfRecords>1</noOfRecords>
      <collections>
         <collection>
            <dateDue>2013-01-08</dateDue>
            <ddRefNo>test1234</ddRefNo>
            <clientRef>test1234</clientRef>
            <name>Account Name</name>
            <sortcode>123456</sortcode>
            <account>12345678</account>
            <amount>101</amount>
            <repeat>Y</repeat>
            <status>SUCCESS</status>
            <failReason />
            <payStatus>PAID</payStatus>
            <paidAmount>101</paidAmount>
            <collectionType>17</collectionType>
            <shortId>SID</shortId>
            <mandateType>L</mandateType>
            <vendor>testVendor</vendor>
            <mandatePayRef>Ref123</mandatePayRef>
         </collection>
      </collections>
   </response>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

Mandate Batch Report

Request

Element Name	Details	Optional	Validation	Character Range	Typical Value
<username>	The username required to identify the user and vendor.	N	Alphanumeric	4-40	Testuser
<command>	The name of the API command.	N			mandateBatchReport
<filters >					
└ **<startDate>**	Starting date of range you wish to filter by. If no date range is given only batches from the current date are returned.	Y	Valid date in the format DD/MM/YYYY		25/10/2013
└ **<endDate >**	Ending date of range you wish to filter by.	Y – Required if the start date has been populated.	Valid date in the format DD/MM/YYYY		29/11/2013
└ **<sortType>**	Field on which the report is to be sorted.	Y – Default is created.	Alpha	1-10	total success
└ **<sortOrder>**	Order in which the report is to be sorted. Ignored if sortType is not provided.	Y – Default is DESC.	Alpha	ASC DESC	ASC DESC
└ **<noOfRows>**	The number of records to be returned by the report. Value must be a multiple of 5	Y – All matching records are required if empty.	Numeric	0-4	20 100

	otherwise it will default to 20.			
<timestamp>	Unix timestamp when the request was made.	N	Numeric	1377791226

Response

Element Name	Description
<status>	Holds the status of the 'mandateBatchReport' request. If the collection action was performed successfully the API returns SUCCESS else returns FAIL.
<errors>	All the errors would be shown in the errors element. This element is empty if the request was successful.
└ **<error>**	
└ **<code>**	Numeric error code.
└ **<detail>**	Human readable error message that describes what the exact error is.
<noOfRecords>	Number of mandate batch records returned as part of the response. If no records are returned the status is SUCCESS and noOfRecords is 0.
<mandateBatches>	
└ **<mandateBatch>**	
└ **<created>**	Batch created date in YYYY-MM-DD HH:MM:SS format.
└ **<sun>**	SUN of the vendor.
└ **<vendorName>**	Short name of the vendor.
└ **<batchname>**	Batch name.
└ **<total>**	Total number.
└ **<success>**	Total success.
└ **<fail>**	Total fail.
└ **<createdby>**	Name of the user who created the batch.
└ **<sentdate>**	Date on which the batch was sent to BACS.
└ **<sentby>**	Name of the user who sent the batch to BACS.

Example

The following is an example of a typical ‘mandateBatchReport’ command:

```
XML REQUEST:

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "apiDtd/mandateBatchReportRequest.dtd">
<ddcm>
   <request>
      <command>mandateBatchReport</command>
      <username>xyzergyd</username>
      <filters>
         <startDate>01/01/2013</startDate>
         <endDate>01/08/2013</endDate>
         <sortType>datetime</sortType>
         <sortOrder>ASC</sortOrder>
         <noOfRows>20</noOfRows>
      </filters>
      <timestamp>0000000000</timestamp>
   </request>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

The following response would be returned:

```
XML RESPONSE:

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM
"https://rsm5.rsmsecure.com/ddcm/apiDtd/mandateBatchReportResponse.dtd">
<ddcm>
   <response>
      <status>SUCCESS</status>
      <errors />
      <noOfRecords>1</noOfRecords>
      <mandateBatches>
         <mandateBatch>
            <created>2013-01-08 16:10:19</created>
            <sun>123456</sun>
            <vendorName>testVendor</vendorName>
            <batchname>123456 MAN 20130108 2</batchname>
            <total>2</total>
            <success>2</success>
            <fail>0</fail>
            <createdby>testUser</createdby>
            <sentdate>2013-01-15</sentdate>
            <sentby>testUser2</sentby>
         </mandateBatch>
      </mandateBatches>
   </response>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

Collection Batch Report

Request

Element Name	Details	Optional	Validation	Character Range	Typical Value
<username>	Required to identify the user and vendor.	N	Alphanumeric	4-40	Testuser
<command>	The name of the API command.	N			collectionBatchReport
<filters>					
└ **<startDate>**	Starting date of range you wish to filter by. If no date range is given only batches from the current date are returned.	Y	Valid date in the format DD/MM/YYYY		25/10/2013
└ **<endDate>**	Ending date of range you wish to filter by.	Y – Required if the start date has been populated.	Valid date in the format DD/MM/YYYY		29/11/2013
└ **<sortType>**	Field on which the report is to be sorted.	Y – Default is created.	Alpha	1-10	total success
└ **<sortOrder>**	Order in which the report is to be sorted. Ignored if sortType is not provided.	Y – Default is DESC.	Alpha	ASC DESC	ASC DESC
└ **<noOfRows>**	The number of records to be returned by the report. Value must be a multiple of 5 otherwise it will	Y – All matching records are required if empty.	Numeric	0-4	20 100

	default to 20.			
<timestamp>	Unix timestamp when the request was made.	N	Numeric	1377791226

Response

Element Name	Description
<status>	Holds the status of the 'collectionBatchReport' request. If the collection action was performed successfully the API returns SUCCESS else returns FAIL.
<errors>	All the errors would be shown in the errors element. This element is empty if the request was successful.
└ **<error>**	
└ **<code>**	Numeric error code.
└ **<detail>**	Human readable error message that describes what the exact error is.
<noOfRecords>	Number of collection batch records returned as part of the response. If no records are returned the status is SUCCESS and noOfRecords is 0.
<collectionBatches>	
└ **<collectionBatch>**	
└ **<created>**	Batch created date in YYYY-MM-DD HH:MM:SS format.
└ **<sun>**	SUN of the vendor.
└ **<vendorName>**	Short name of the vendor.
└ **<batchName>**	Batch name.
└ **<collectDate>**	Collection date in YYYY-MM-DD format.
└ **<total>**	Total number of collections part of the batch.
└ **<success>**	Total successful collections part of the batch.
└ **<successAmount>**	Total collection amount that was successful.
└ **<fail>**	Total fail collections part of the batch.
└ **<failAmount>**	Total collection amount that was failed.
└ **<createdby>**	Name of the user who created the batch.
└ **<sentdate>**	Date on which the batch was sent to BACS.
└ **<sentby>**	Name of the user who sent the batch to BACS.

Example

The following is an example of a typical ‘collectionBatchReport’ command:

XML REQUEST:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "apiDtd/collectionBatchReportRequest.dtd">
<ddcm>
   <request>
      <command>collectionBatchReport</command>
      <username>xyzergyd</username>
      <filters>
         <startDate>01/01/2013</startDate>
         <endDate>01/08/2013</endDate>
         <sortType>datetime</sortType>
         <sortOrder>ASC</sortOrder>
         <noOfRows>20</noOfRows>
      </filters>
      <timestamp>0000000000</timestamp>
   </request>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

The following response would be returned:

XML RESPONSE:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM
"https://rsm5.rsmsecure.com/ddcm/apiDtd/collectionBatchReportResponse.dtd">
<ddcm>
   <response>
      <status>SUCCESS</status>
      <errors />
      <noOfRecords>1</noOfRecords>
      <collectionBatches>
         <collectionBatch>
            <created>2013-01-08 16:10:26</created>
            <sun>123456</sun>
            <clientRef>test1234</clientRef>
            <vendorName>testVendor</vendorName>
            <batchName>123456 COL 20130108_05 3 £301.50</batchName>
            <collectDate>2013-01-05</collectDate>
            <total>3</total>
            <success>2</success>
            <successAmount>201.00</successAmount>
            <fail>1</fail>
            <failAmount>100.50</failAmount>
            <createdby>testUser</createdby>
            <sentdate>2013-01-15</sentdate>
            <sentby>testUser2</sentby>
         </collectionBatch>
      </collectionBatches>
   </response>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

Letter Batch Report

Request

Element Name	Details	Optional	Validation	Character Range	Typical Value
<username>	The username required to identify the user and vendor.	N	Alphanumeric	4-40	Testuser
<command>	The name of the API command.	N			letterBatchReport
<filters>					
└ **<sortType>**	Field on which the report is to be sorted.	Y – Default is created.	Alpha	1-10	total sentBy
└ **<sortOrder>**	Order in which the report is to be sorted. Ignored if sortType is not provided.	Y – Default is DESC.	Alpha	ASC DESC	ASC DESC
└ **<noOfRows>**	The number of records to be returned by the report. Value must be a multiple of 5 otherwise it will default to 20.	Y – Default is blank, which would retrieve all the rows returned by the search.	Numeric	0-4	20 100
<timestamp>	Unix timestamp when the request was made.	N	Numeric		1377791226

Response

Node/Element	Description
<status>	Holds the status of the 'letterBatchReport' request. If the action was performed successfully the API returns SUCCESS otherwise FAIL is returned.
<errors>	All the errors would be shown in the errors element. This element is empty if the request was successful.
└ **<error>**	
└ **<code>**	Numeric error code.
└ **<detail>**	Human readable error message that describes what the exact error is.
<noOfRecords>	Number of letter batch records returned as part of the response. If no records are returned the status is SUCCESS and noOfRecords is 0.
<letterBatches>	
└ **<letterBatch>**	
└ **<created>**	Batch created date in YYYY-MM-DD HH:MM:SS format.
└ **<sun>**	The service user number of the vendor.
└ **<vendorName>**	Short name of the vendor.
└ **<batchname>**	The name of the letter batch.
└ **<total>**	The number of mandates included in the batch.
└ **<createdby>**	Name of the user who created the batch.
└ **<sentdate>**	Date on which the letter batch was marked as sent.
└ **<sentby>**	Name of the user who marked the letter batches as sent.

Example

The following is an example of a typical ‘letterBatchReport’ command:

XML REQUEST:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "apiDtd/letterBatchReportRequest.dtd">
<ddcm>
   <request>
      <command>letterBatchReport</command>
      <username>xyzergyd</username>
      <filters>
         <sortType>datetime</sortType>
         <sortOrder>ASC</sortOrder>
         <noOfRows>20</noOfRows>
      </filters>
      <timestamp>0000000000</timestamp>
   </request>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

The following response would be returned:

```
XML RESPONSE:

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM
"https://rsm5.rsmsecure.com/ddcm/apiDtd/letterBatchReportResponse.dtd">
<ddcm>
   <response>
      <status>SUCCESS</status>
      <errors />
      <noOfRecords>1</noOfRecords>
      <letterBatches>
         <letterBatch>
            <created>2013-01-08</created>
            <sun>123456</sun>
            <vendorName>testVendor</vendorName>
            <batchname>837955 CONFIRM 20110824 5</batchname>
            <total>3</total>
            <createdby>testUser</createdby>
            <sentdate>2013-01-15</sentdate>
            <sentby>testUser2</sentby>
         </letterBatch>
      </letterBatches>
   </response>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

Collection Statement

Request

Element Name	Details	Optional	Validation	Character Range	Typical Value
<username>	The username required to identify the user and vendor.	N	Alphanumeric	4-40	Testuser
<command>	The name of the API command.	N			collectionStatement
<filters>					
└ **<collectionDate>**	The date in which collection was made.	Y – Statements can be filtered by either the collection date or the statement date.	Valid date in the format DD/MM/YYYY		25/10/2013
└ **<statementDate>**	The date in which the statement was generated.	Y – Statements can be filtered by either the collection date or the statement date.	Valid date in the format DD/MM/YYYY		29/11/2013
<timestamp>	Unix timestamp when the request was made.	N	Numeric		1377791226

Response

Node/Element	Description
<status>	Holds the status of the 'collectionStatement' request. If the action was performed successfully the API returns SUCCESS otherwise FAIL is returned.
<errors>	All the errors would be shown in the errors element. This element is empty if the request was successful.
└ **<error>**	
└ **<code>**	Numeric error code.
└ **<detail>**	Human readable error message that describes what the exact error is.
<collectionStatement>	
└ **<header>**	This is the name of the collection statement
└ **<collections>**	
└ **<collection>**	
└ **<vendor>**	Vendor's short name. **Only returned for supervendors.**
└ **<ddRefNo>**	BACS Direct Debit reference number.
└ **<clientRef>**	Client reference.
└ **<name>**	Account holder's name.
└ **<reason>**	Collection fail reason.
└ **<amount>**	Amount collected.
└ **<mandatePayRef>**	The pay ref of the collections mandate when the collection was created.
└ **<collectionTotal>**	Holds a summary of the collections made in this statement.
└ **<totalCollected>**	The total amount raised by all collections.
└ **<totalItemsSent>**	The total number of collections sent.
└ **<charges>**	**Only shown if monthly statements are not enabled.**

└ **<mandates>**	Holds information on all mandate setup and charges made since the last statement. **This element is only returned if the vendor is enabled for itemised statements.**
└ **<setups>**	Contains a list of mandates which were setup since the last statement.
└ **<mandate>**	
└ **<vendor>**	Vendor’s short name. **Only returned for supervendors.**
└ **<ddRefNo>**	BACS Direct Debit reference number.
└ **<clientRef>**	Client reference.
└ **<name>**	Account holder’s name.
└ **<addmandates>**	Contains a list of mandates which were submitted manually through DDCM since the last statement.
└ **<mandate>**	
└ **<vendor>**	Vendor’s short name. **Only returned for supervendors.**
└ **<ddRefNo>**	BACS Direct Debit reference number.
└ **<clientRef>**	Client reference.
└ **<name>**	Account holder’s name.
└**<amendments>**	Contains a list of mandates which were amended since the last statement.
└ **<mandate>**	
└ **<vendor>**	Vendor’s short name. **Only returned for supervendors.**
└ **<ddRefNo>**	BACS Direct Debit reference number.
└ **<clientRef>**	Client reference.
└ **<name>**	Account holder’s name.
└ **<letters>**	Contains a list of mandates that were sent confirmation letters since the last statement.
└ **<mandate>**	

└ **<vendor>**	Vendor's short name. **Only returned for supervendors.**
└ **<ddRefNo>**	BACS Direct Debit reference number.
└ **<clientRef>**	Client reference.
└ **<name>**	Account holder's name.
└ **<emails>**	Contains a list of mandates that were sent confirmation emails since the last statement.
└ **<mandate>**	
└ **<vendor>**	Vendor's short name. **Only returned for supervendors.**
└ **<ddRefNo>**	BACS Direct Debit reference number.
└ **<clientRef>**	Client reference.
└ **<name>**	Account holder's name.
└ **<balanceBeforeCharges>**	Holds a vendors total balance before any charges have been applied.
└ **<amount>**	The vendors total balance before any charges have been applied.
└ **<mandateSetupsAndTransmissions>**	Holds the total charges applied for mandate setups and transmissions.
└ **<number>**	Number of setups and transmissions.
└ **<amount>**	Total cost of all the setups and transmissions.
└ **<addMandateOnscreen>**	Holds the total charges applied for adding mandates through DDCM.
└ **<number>**	Total number of added mandates.
└ **<amount>**	Total cost of adding mandates.
└ **<mandateAmendmentsAndTransmissions>**	Holds the total charges applied for mandate amendments and transmissions.
└ **<number>**	The total number of amended mandates.
└ **<amount>**	The total cost of amending mandates.
└ **<collectionFiles>**	Holds total charges applied for generating collection

	files.
└ **<number>**	Total number of collection files generated since the last statement.
└ **<amount>**	Total cost of generating those collection files.
└ **<collectionItems>**	
└ **<number>**	
└ **<amount>**	
└ **<collectionItemsAtMinCharge>**	
└ **<number>**	
└ **<amount>**	
└ **<maxTransactionValueSurcharge>**	
└ **<number>**	
└ **<amount>**	
└ **<minTransactionValueSurcharge>**	
└ **<number>**	
└ **<amount>**	
└ **<unpaidDds>**	
└ **<number>**	
└ **<amount>**	
└ **<indemnityCharge>**	
└ **<number>**	
└ **<amount>**	
└ **<confirmationLetter>**	Holds total charges applied for generating confirmation letters. **Only if vendor has confirmation letters enabled.**
└ **<number>**	Total number of letters generated.

└ **<amount>**	Total amount charged for generating those letters.
└ **<confirmationEmails>**	Holds total charges applied for generating confirmation email. **Only if vendor has confirmation emails enabled.**
└ **<number>**	Total number of emails generated.
└ **<amount>**	Total amount charged for generating those letters.
└ **<monthlyServiceCharge>**	
└ **<number>**	
└ **<amount>**	
└ **<totals>**	Holds totals of all charges applied. **Only shown if monthly statements is set as No**
└ **<totalCharges>**	Total amount of charges applied
└ **<amount>**	
└ **<vat>**	Total amount of VAT applied.
└ **<amount>**	
└ **<totalChargesIncVat>**	Total amount for charges including VAT
└ **<amount>**	
└ **<balancePayable>**	Total amount collected minus all charges.
└ **<amount>**	

EXAMPLE

The following is an example of a typical ‘collectionStatement’ command:

XML REQUEST:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "apiDtd/letterBatchReportRequest.dtd">
<ddcm>
   <request>
      <command>collectionStatement</command>
      <username>xyzergyd</username>
      <filters>
         <collectionDate>01/01/2013</collectionDate>
      </filters>
      <timestamp>0000000000</timestamp>
   </request>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

The following response would be returned:

XML RESPONSE:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "https://rsm5.rsmsecure.com/ddcm/apiDtd/
collectionStatementResponse.dtd ">
<ddcm>
   <response>
      <status>SUCCESS</status>
      <errors />
      <collectionStatement>
         <header>RSMDD_Statement_blah _839144_2013-07-25</header>
         <collections>
            <collection>
               <vendor>03</vendor>
               <ddRefNo>test1234</ddRefNo>
               <clientRef>test1234</clientRef>
               <name>Account Name</name>
               <reason>123456</reason>
               <amount>101.12</amount>
               <mandatePayRef>mandateref123</mandatePayRef>
            </collection>
         </collections>
         <collectionTotal>
            <totalCollected>1000</totalCollected>
            <totalItemsSent>1000</totalItemsSent>
         </collectionTotal>
         <reclaims>
            <reclaim>
               <ddRefNo>test1234</ddRefNo>
               <clientRef>test1234</clientRef>
               <name>Account Name</name>
               <reason>123456</reason>
               <amount>101.12</amount>
               <vendor>03</vendor>
            </reclaim>
         </reclaims>
         <reclaimTotal>
            <totalReclaimed>1000</totalReclaimed>
         </reclaimTotal>
         <charges>
            <mandates>
               <setups>
                  <mandate>
                     <ddRefNo>test1234</ddRefNo>
                     <clientRef>test1234</clientRef>
                     <name>Account Name</name>
                     <vendor>03</vendor>
                  </mandate>
               </setups>
               <addmandates>
                  <mandate>
                     <ddRefNo>test1234</ddRefNo>
                     <clientRef>test1234</clientRef>
                     <name>Account Name</name>
                     <vendor>03</vendor>
                  </mandate>
               </addmandates>
               <ammendments>
                  <mandate>
                     <ddRefNo>test1234</ddRefNo>
                     <clientRef>test1234</clientRef>
                     <name>Account Name</name>
                     <vendor>03</vendor>
                  </mandate>
               </ammendments>
               <letters>
                  <mandate>
                     <ddRefNo>test1234</ddRefNo>
                     <clientRef>test1234</clientRef>
                     <name>Account Name</name>
                     <vendor>03</vendor>
                  </mandate>
               </letters>
               <emails>
                  <mandate>
                     <ddRefNo>test1234</ddRefNo>
                     <clientRef>test1234</clientRef>
                     <name>Account Name</name>
                     <vendor>03</vendor>
                  </mandate>
               </emails>
            </mandates>
            <balanceBeforeCharges>
               <amount>12.00</amount>
            </balanceBeforeCharges>
            <mandateSetupsAndTransmissions>
               <number>12</number>
               <amount>12.00</amount>
            </mandateSetupsAndTransmissions>
            <addMandateOnscreen>
               <number>12</number>
               <amount>12.00</amount>
            </addMandateOnscreen>
            <mandateAmendmentsAndTransmissions>
               <number>12</number>
               <amount>12.00</amount>
            </mandateAmendmentsAndTransmissions>
            <collectionFiles>
               <number>12</number>
               <amount>12.00</amount>
            </collectionFiles>
            <collectionItems>
               <number>12</number>
               <amount>12.00</amount>
            </collectionItems>
            <collectionItemsAtMinCharge>
```

```
               <number>12</number>
               <amount>12.00</amount>
            </collectionItemsAtMinCharge>
            <maxTransactionValueSurcharge>
               <number>12</number>
               <amount>12.00</amount>
            </maxTransactionValueSurcharge>
            <minTransactionValueSurcharge>
               <number>12</number>
               <amount>12.00</amount>
            </minTransactionValueSurcharge>
            <unpaidDds>
               <number>12</number>
               <amount>12.00</amount>
            </unpaidDds>
            <indemnityCharge>
               <number>12</number>
               <amount>12.00</amount>
            </indemnityCharge>
            <confirmationLetter>
               <number>12</number>
               <amount>12.00</amount>
            </confirmationLetter>
            <confirmationEmails>
               <number>12</number>
               <amount>12.00</amount>
            </confirmationEmails>
            <monthlyServiceCharge>
               <number>12</number>
               <amount>12.00</amount>
            </monthlyServiceCharge>
         </charges>
         <totals>
            <totalCharges>
               <amount>12.00</amount>
            </totalCharges>
            <vat>
               <amount>12.00</amount>
            </vat>
            <totalChargesIncVat>
               <amount>12.00</amount>
            </totalChargesIncVat>
            <balancePayable>
               <amount>12.00</amount>
            </balancePayable>
         </totals>
      </collectionStatement>
   </response>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

Monthly Statement

Request

Element Name	Details	Optional	Validation	Character Range	Typical Value
<username>	The username required to identify the user and vendor.	N	Alphanumeric	4-40	Testuser
<command>	The name of the API command.	N			monthlyStatement
<filters>					
└ **<monthYear>**	The month and year of the requested statement.	N	Valid date in the format MM/YYYY		10/2013
<timestamp>	Unix timestamp when the request was made.	N	Numeric		1377791226

Response

Node/Element	Description
<status>	Holds the status of the 'monthlyStatement' request. If the action was performed successfully the API returns SUCCESS otherwise FAIL is returned.
<errors>	All the errors would be shown in the errors element. This element is empty if the request was successful.
└ **<error>**	
└ **<code>**	Numeric error code.
└ **<detail>**	Human readable error message that describes what the exact error is.
< monthlyStatement>	
└ **<header>**	This is the name of the collection statement.
└ **<charges>**	**Only shown if monthly statements are disabled.**
└ **<mandates>**	Holds information on all mandate setup and charges made since the last statement. **This element is only returned if the vendor is enabled for itemised statements.**
└ **<setups>**	Contains a list of mandates which were setup since the last statement.
└ **<mandate>**	
└ **<vendor>**	Vendor's short name. **Only returned for supervendors.**
└ **<ddRefNo>**	BACS Direct Debit reference number.
└ **<clientRef>**	Client reference.
└ **<name>**	Account holder's name.
└ **<addmandates>**	Contains a list of mandates which were added on screen since the last statement.
└ **<mandate>**	
└ **<vendor>**	Vendor's short name. **Only returned for supervendors.**

└ **<ddRefNo>**	BACS Direct Debit reference number.
└ **<clientRef>**	Client reference.
└ **<name>**	Account holder’s name.
└**<amendments>**	Contains a list of mandates which were amended since the last statement.
└ **<mandate>**	
└ **<vendor>**	Vendor’s short name. **Only returned for supervendors.**
└ **<ddRefNo>**	BACS Direct Debit reference number.
└ **<clientRef>**	Client reference.
└ **<name>**	Account holder’s name.
└ **<letters>**	Contains a list of mandates that were sent confirmation letters since the last statement.
└ **<mandate>**	
└ **<vendor>**	Vendor’s short name. **Only returned for supervendors.**
└ **<ddRefNo>**	BACS Direct Debit reference number.
└ **<clientRef>**	Client reference.
└ **<name>**	Account holder’s name.
└ **<emails>**	Contains a list of mandates that were sent confirmation emails since the last statement.
└ **<mandate>**	
└ **<vendor>**	Vendor’s short name. **Only returned for supervendors.**
└ **<ddRefNo>**	BACS Direct Debit reference number.
└ **<clientRef>**	Client reference.
└ **<name>**	Account holder’s name.
└ **<balanceBeforeCharges>**	Holds a vendors total balance before any charges have been applied.
└ **<amount>**	The vendors total balance before any charges have

	been applied.
└ **<mandateSetupsAndTransmissions>**	Holds the total charges applied for mandate setups and transmissions.
└ **<number>**	Number of setups and transmissions.
└ **<amount>**	Total cost of all the setups and transmissions.
└ **<addMandateOnscreen>**	Holds the total charges applied for adding mandates through DDCM.
└ **<number>**	Total number of added mandates.
└ **<amount>**	Total cost of adding mandates.
└ **<mandateAmendmentsAndTransmissions>**	Holds the total charges applied for mandate amendments and transmissions.
└ **<number>**	The total number of amended mandates.
└ **<amount>**	The total cost of amending mandates.
└ **<collectionFiles>**	Holds total charges applied for generating collection files.
└ **<number>**	Total number of collection files generated since the last statement.
└ **<amount>**	Total cost of generating those collection files.
└ **<collectionItems>**	
└ **<number>**	
└ **<amount>**	
└ **<collectionItemsAtMinCharge>**	
└ **<number>**	
└ **<amount>**	
└ **<maxTransactionValueSurcharge>**	
└ **<number>**	
└ **<amount>**	

└ <minTransactionValueSurcharge>	
└ <number>	
└ <amount>	
└ <unpaidDds>	
└ <number>	
└ <amount>	
└ <indemnityCharge>	
└ <number>	
└ <amount>	
└ <confirmationLetter>	Holds total charges applied for generating confirmation letters. **Only if vendor has confirmation letters enabled.**
└ <number>	Total number of letters generated.
└ <amount>	Total amount charged for generating those letters.
└ <confirmationEmails>	Holds total charges applied for generating confirmation email. **Only if vendor has confirmation emails enabled.**
└ <number>	Total number of emails generated.
└ <amount>	Total amount charged for generating those letters.
└ <monthlyServiceCharge>	
└ <number>	
└ <amount>	
└ <totals>	Holds totals of all charges applied. **Only shown if monthly statements are disabled.**
└ <totalCharges>	Total amount of charges applied
└ <amount>	
└ <vat>	Total amount of VAT applied.

└ **<amount>**	
└ **<totalChargesIncVat>**	Total amount for charges including VAT.
└ **<amount>**	
└ **<balancePayable>**	Total amount collected minus all charges.
└ **<amount>**	

EXAMPLE

The following is an example of a typical ‘monthlyStatement’ command:

XML REQUEST:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "apiDtd/monthlyStatementRequest.dtd">
<ddcm>
   <request>
      <command>monthlyStatement</command>
      <username>xyzergyd</username>
      <filters>
         <monthYear>10/2013</monthYear>
      </filters>
      <timestamp>0000000000</timestamp>
   </request>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

The following response would be returned:

XML RESPONSE:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM
"https://rsm5.rsmsecure.com/ddcm/apiDtd/monthlyStatementResponse.dtd">
<ddcm>
   <response>
      <status>SUCCESS</status>
      <errors />
      <monthlyStatement>
         <header>RSMDD_Statement_blah _839144_2013-07-25</header>
         <charges>
            <mandates>
               <setups>
                  <mandate>
                     <ddRefNo>test1234</ddRefNo>
                     <clientRef>test1234</clientRef>
                     <name>Account Name</name>
                     <vendor>03</vendor>
                  </mandate>
               </setups>
               <addmandates>
                  <mandate>
                     <ddRefNo>test1234</ddRefNo>
                     <clientRef>test1234</clientRef>
                     <name>Account Name</name>
                     <vendor>03</vendor>
                  </mandate>
               </addmandates>
               <ammendments>
                  <mandate>
                     <ddRefNo>test1234</ddRefNo>
                     <clientRef>test1234</clientRef>
                     <name>Account Name</name>
                     <vendor>03</vendor>
                  </mandate>
               </ammendments>
               <letters>
                  <mandate>
                     <ddRefNo>test1234</ddRefNo>
                     <clientRef>test1234</clientRef>
                     <name>Account Name</name>
                     <vendor>03</vendor>
                  </mandate>
               </letters>
               <emails>
                  <mandate>
                     <ddRefNo>test1234</ddRefNo>
                     <clientRef>test1234</clientRef>
                     <name>Account Name</name>
                     <vendor>03</vendor>
                  </mandate>
               </emails>
            </mandates>
            <balanceBeforeCharges>
               <amount>12.00</amount>
            </balanceBeforeCharges>
            <mandateSetupsAndTransmissions>
               <number>12</number>
               <amount>12.00</amount>
            </mandateSetupsAndTransmissions>
            <addMandateOnscreen>
               <number>12</number>
               <amount>12.00</amount>
            </addMandateOnscreen>
            <mandateAmendmentsAndTransmissions>
               <number>12</number>
               <amount>12.00</amount>
            </mandateAmendmentsAndTransmissions>
            <collectionFiles>
               <number>12</number>
               <amount>12.00</amount>
            </collectionFiles>
            <collectionItems>
               <number>12</number>
               <amount>12.00</amount>
            </collectionItems>
            <collectionItemsAtMinCharge>
               <number>12</number>
               <amount>12.00</amount>
            </collectionItemsAtMinCharge>
            <maxTransactionValueSurcharge>
               <number>12</number>
               <amount>12.00</amount>
            </maxTransactionValueSurcharge>
            <minTransactionValueSurcharge>
               <number>12</number>
               <amount>12.00</amount>
            </minTransactionValueSurcharge>
            <unpaidDds>
               <number>12</number>
               <amount>12.00</amount>
            </unpaidDds>
            <indemnityCharge>
               <number>12</number>
               <amount>12.00</amount>
            </indemnityCharge>
            <confirmationLetters>
               <number>12</number>
               <amount>12.00</amount>
            </confirmationLetters>
            <confirmationEmails>
               <number>12</number>
               <amount>12.00</amount>
            </confirmationEmails>
            <monthlyServiceCharge>
```

```
               <number>12</number>
               <amount>12.00</amount>
            </monthlyServiceCharge>
         </charges>
         <totals>
            <totalCharges>
               <amount>12.00</amount>
            </totalCharges>
            <vat>
               <amount>12.00</amount>
            </vat>
            <totalChargesIncVat>
               <amount>12.00</amount>
            </totalChargesIncVat>
            <balancePayable>
               <amount>12.00</amount>
            </balancePayable>
         </totals>
      </monthlyStatement>
   </response>
   <signature>abcdefghijklmnopqrstuvwxyz</signature>
</ddcm>
```

Response Status Check

Used to retrieve the response of a previously executed command using it's generated signature. 'Response Status Check' will only work using signatures from the following commands: setMandates & setCollections.

Request

Element Name	Details	Optional	Validation	Character Range	Typical Value
<command>	The name of the API command.	N			responseStatusCheck
<username>	The username required to identify the user and vendor.	N	Alphanumeric	4-40	Testuser
<previousSignature>	The signature of the original command.	N	Alphanumeric	64	
<timestamp>	Unix timestamp when the request was made.	N	Numeric		1377791226

Response

An error response is returned if the previous signature value is:

- Not sent.
- Doesn’t match with any previous requests.
- If the previous request was also a ‘responseStatusCheck’ command.

In all other cases this command gives out the response which was given out after the previous command was executed.

EXAMPLE

The following is an example of a typical ‘responseStatusCheck’ command:

XML REQUEST:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ddcm SYSTEM "apiDtd/responseStatusCheckRequest.dtd">
<ddcm>
   <request>
      <command>responseStatusCheck</command>
      <username>username</username>
      <previousSignature />
      <timestamp>0000000000</timestamp>
   </request>
   <signature />
</ddcm>
```

APPENDIX A: ERROR CODES

Error Code	Related Element	Message	Error Could Occur For The Following Reasons
1001		Invalid request	Request doesn't match the supplied DTD.
1002		Invalid response	There was a problem returning the response. Please contact RSM Support.
1003		An error occurred	An error while trying to process the request. Please contact RSM Support.
1004		Given Signature does not match our generated one	The signature sent as part of the request does not match with the one generated by DDCM.
1005		This is not a unique request	
2001	username	Username is required	Username field is blank.
2002	username	User record could not be found	Username may have been typed incorrectly or user has not been setup to use DDCM API.
2011	Command	Command is required	Command field is required.
2012	Command	Command not recognised	Given command is not valid.
2013	Command	User does not have access to this feature	User doesn't have the access levels required to use this command.
2014	Command	No mandate data sent as part of the request	Mandates element is empty or not provided as part of the 'setMandates' request.
2015	Command	No collection data sent as part of the request	Collections element is empty or not provided as part of the 'setCollections' request.
2016	Command	Unable to fetch the data	DDCM could not read data from the request. Please contact RSM support for assistance.
2017	Command	Headers not defined for type	DDCM could not read data from the request. Please contact RSM support

			for assistance.
2018	Username	Vendor record could not be found	Vendor data associated to the given user not found.
2021	Trading Name	Trading name is malformed or invalid	
2031	Contact Name	Last Name is malformed or invalid	
2041	Address1	Address1 is malformed or invalid	
2042	Address1	Address1 field not supplied	Address1 is mandatory if the vendor has address checking enabled.
2051	Address2	Address2 is malformed or invalid	
2061	Address3	Address3 is malformed or invalid	
2071	Town	Town is malformed or invalid	
2072	Town	Town field is not supplied	Town is mandatory if the vendor has address checking enabled.
2081	Postcode	Postcode is malformed or invalid	
2082	Postcode	Postcode field is not supplied	Postcode is mandatory if the vendor has address checking enabled.
2091	Phone	Telephone is malformed or invalid	
2101	Email	Email is malformed or invalid	
2102	Email	Email is not valid or not supplied	Email is mandatory if vendor has ‘confirm email’ set as mandatory.
2111	ClientRef	Client Reference is malformed or invalid	
2112	ClientRef	Active Mandate with that ClientRef already exists	A mandate with provided ClientRef and status as LIVE, AMEND, NEW, PENDING or RESUBMIT already

			exists.
2113	ClientRef	Client Reference not supplied	ClientRef is mandatory for this command.
2114	ClientRef	Invalid ClientRef - Duplicate ClientRef within request	The request has more than one mandate elements for same ClientRef.
2115	ClientRef	Invalid ClientRef	The given client ref does not match against the mandate details in DDCM.
2121	Account Name	Account holder name is malformed or invalid	
2122	Account Name	Account Name is not supplied	
2131	Account Number	Account number is malformed or invalid	
2132	Account Number	Account number not supplied	
2133	Account Number	Account number invalid length	Account number has to 8 digits.
2134	Account Number	Account number contains invalid characters	Account number must be numeric.
2141	Sort Code	Sort code is malformed or invalid	Sort code is not numeric.
2142	Sort Code	Sort code not supplied	
2143	Sort Code	Sort code invalid length	Sort code has to 6 digits.
2144	Sort Code	Sort code contains invalid characters	Sort code must be numeric.
2151	Action	Invalid record type	Action is invalid.
2152	Action	Mandate cancels are not set for this Vendor	
2153	Action	Collection cannot be deleted	Only collections with a status of NEW can be deleted.
2154	Action	The collection is not editable	Only future collections with a status of NEW can be edited.

2155	Action	Valid collection does not exist	Collection with given details does not exists in the database (for editing or deletion).
2161	DDRefNo	DDReference Number is malformed or invalid	DDReference number is not valid.
2162	DDRefNo	Invalid DDRefNo - A reference was supplied but Vendor settings do not allow this	The vendor is set to generate DDRefNo automatically, but it was still supplied as part of request.
2163	DDRefNo	Invalid DDRefNo - Contains invalid characters	DDRefNo contains chars other than a-z, A-Z, 0-9, -, . or /
2164	DDRefNo	Invalid DDRefNo - Requires <ddreflength> characters with at least 6 uppercase alpha & numeric	DDRefNo should contain at least 6 alphanumeric characters.
2165	DDRefNo	Invalid DDRefNo - Length must be <ddreflength> characters	DDRefNo length does not match with the vendor settings.
2166	DDRefNo	Valid Mandate does not exist	There are no mandates with the provided DDRefNo which have a status of LIVE, FAILED or CANCELLED.
2167	DDRefNo	DDRefNo not supplied	
2168	DDRefNo	Invalid DDRefNo - Duplicate DDRefNo within request	The request has more than one mandate elements with the same DDRefNo.
2169	DDRefNo	Mandate already exists	An active mandate with that DDRefNo already exists.
2171	Amount	Amount is malformed or invalid	
2172	Amount	Invalid collection amount	
2173	Amount	Amount not supplied	
2174	Amount	Amount cannot be greater than <maxcollectionamount>	Mandate’s amount if supplied cannot be greater than the vendors set maximum collection amount.

2175	Amount	Amount limit exceeded in payments dated <problem dates>	Mandate or collection amount in request exceeds the vendors set daily limit.
2181	Frequency	Payment Frequency is malformed or invalid	
2182	Frequency	Repeating mandates not allowed	Vendor has not been setup to allow repeating mandates.
2183	Frequency	Non-repeating mandates not allowed	Vendor has been setup to only allow repeating mandates, a payment frequency must be supplied.
2184	Frequency	Automatic repeat collections were either not opted or not allowed	Frequency or amount is supplied for non-repeating mandate.
2185	Frequency	Payment frequency must not be supplied for non repeating mandates	
2186	Frequency	Invalid payment frequency	Frequency must be either: 1,3,6, 12 or monthly, quarterly, half yearly, yearly.
2187	Frequency	Payment frequency not supplied	Frequency is mandatory for a repeating mandate.
2191	Start Date	Date is malformed or invalid	
2192	Start Date	Invalid start date	
2193	Start Date	Start Date must be at least 5 working days in the future	
2194	Start Date	Start Date must not be later than 13 months in the future	
2195	Date	Invalid collection date	
2196	Date	Collection within 3 working days	
2197	Date	Collection within 5 working days of mandate lodged	Collection date must be later than 5 working days if the mandate is not LIVE.

2198	Start Date	Start Date must be at least 3 working days in the future	
2201	Mandate Type	Mandate type not supplied	The vendor is set to use mandate types, but it was not supplied in the request.
2202	Mandate Type	Mandate type is not valid	Mandate type is supplied but is not valid.
2211	Short ID	Short ID not provided	Short id is not provided, it is mandatory for super vendors.
2212	Short ID	Invalid short id	Short id provided does not match with any of the vendor’s sub vendors.
2221	End Date	End Date is malformed or invalid	
2222	End Date	End date must be later than the start date	
2223	End Date	Invalid end date	
2231	Account Errors		Error messages returned by validate (account check service).
2241	QAS Errors	Quick address check errors	
2251	SUN	SUM is malformed or invalid	SUN has to be a 6 digit numeric value.
2252	SUN	Invalid Service User Number	SUN supplied does not match with the vendor record.
2261	Status	Invalid status supplied	Status value supplied as part of the report filter is not valid.
2262	Sort Type	Invalid sort type supplied	Sort type value supplied as part of the report filter is not valid.
2263	Sort Order	Invalid sort order supplied	Sort order value supplied as part of the report filter is not valid.
2271	Existing Amount	Invalid existing collection amount	
2272	Existing Amount	Existing amount is malformed or invalid	

2281	Existing Date	Invalid existing collection date	
2282	Existing Date	Existing date is malformed or invalid	
2291	Collection or Statement Dates	Collection or statement date not supplied	Neither a collection date nor a statement date was supplied as part of the ‘collectionStatement’ request.
2292	Collection Date	Invalid collection date	Collection date sent as part of the ‘collectionStatement’ request is invalid.
2293	Statement Date	Invalid statement date	Statement date sent as part of the ‘collectionStatement’ request is invalid.
2294	Statement	No statement found in the search	
2295	Month Year	Statement date not supplied	Statement date (month year) is not supplied as part of monthly statement request.
2296	Vendor Settings	Monthly statements not enabled for the vendor	
2297	Statement	Problem fetching the statement	
2301	Response Check	Record with given signature could not be found	
2302	Response Check	Unable to load API response due to an invalid signature	
2303	Response Check	Unable to load API response due to a invalid vendor ID	
2304	Response Check	The given signature relates to a responseStatusCheck command	The original command associated with the given signature is ‘responseCheck’. The API does not do a ‘responseCheck’ for a ‘responseCheck’.
2311	Payment Reference	Payment reference is malformed or invalid	