Bank Bill Payment Service Specifications

Version 1.3

Table of Contents

Document Control

Version	Owner	Date	Reviewed	Date	Change
1.0	Mohamed Aboulzahab	15/01/2018	Amr Khater	06/10/2010	Initial Version.
1.1	Ahmed Hamdy	26/02/2018	Mohamed Aboulzahab	27/02/2017	Add CancelPayment message
1.2	Ahmed Hamdy	14/03/2018	Mohamed Aboulzahab	15/03/2018	Edit cancel payment to have PmtRec and cancelReason
1.3	Ahmed Hamdy	16/12/2019	Mohamed Aboulzahab	17/12/2019	Add BillingAcc and BillNo to inquiry status

Bill Information Acquisition / Payment

1. Introduction

Bill Payment Service is a generic Electronic Bill Presentment and Payment (EBPP) service that enable different parties to share a common infrastructure that enabled the Billers to integrate with the system for bill presentment, and the Banks for bill enquiry and payment.

The Business Processes that are provided in this document are intended to allow the Billers and the Banks to understand the scope of Bill Payment Service integration and to identify their roles and responsibilities in the implementation of the processes, while the technical part that reflects specification of exchanged messages will help both parties to build their infrastructure to integrate with the Bill Payment Service portal.

2. Business Process

2.1. Process Overview

The Bill Payment Service Business Processes for EBPP from the Billers' perspective are explained in this section. Participating Billers are required to implement their part of the processes according to these specifications.

Billers will upload bill summary data to Bill Payment Service on a regular schedule that gets stored in the Bill Payment Service database. Bill Payment Service will consolidate bill summary data from the various Billers and present it to the relevant bill payer upon a request initiated from the Banks' channels. It is important to note that it will be Bill Payment Service, not the Banks, which will store bill data. Bill Payment Service will also apply business rules, if required by the Biller, before a payment is accepted based on the bill uploaded information.

The channels of the Banks will interact with Bill Payment Service in real-time through the Banks' back-end systems for all customer-facing scenarios or through the service portal. The Banks' systems will be responsible for gathering customer requests such as bill and payment inquiries from their channels and sending them to Bill Payment Service for processing. In response, the Banks will receive bill summary data from Bill Payment Service that have to be presented to the customer on the channels.

Customers may follow a query on their bills with the payment of single or multiple bills. The Banks' systems will capture the payment information entered by the Customers and forward it to Bill Payment Service for validation. Bill Payment Service will apply business rules and determine if the payment initialized by the Customers is valid. Bill Payment Service will send back the results of the validation to the Banks. If the payment is found to be valid, the Bank will collect payment from the Customers. Payment collection will be done either by debiting the Customer's account or by collecting the cash, depending on the channel type.

At the end of each day, the Bill Payment Service will initiate the reconciliation process by sending summary and detailed reports of that day's transactions to all parties to be used for internal reconciliation and settlement matching.

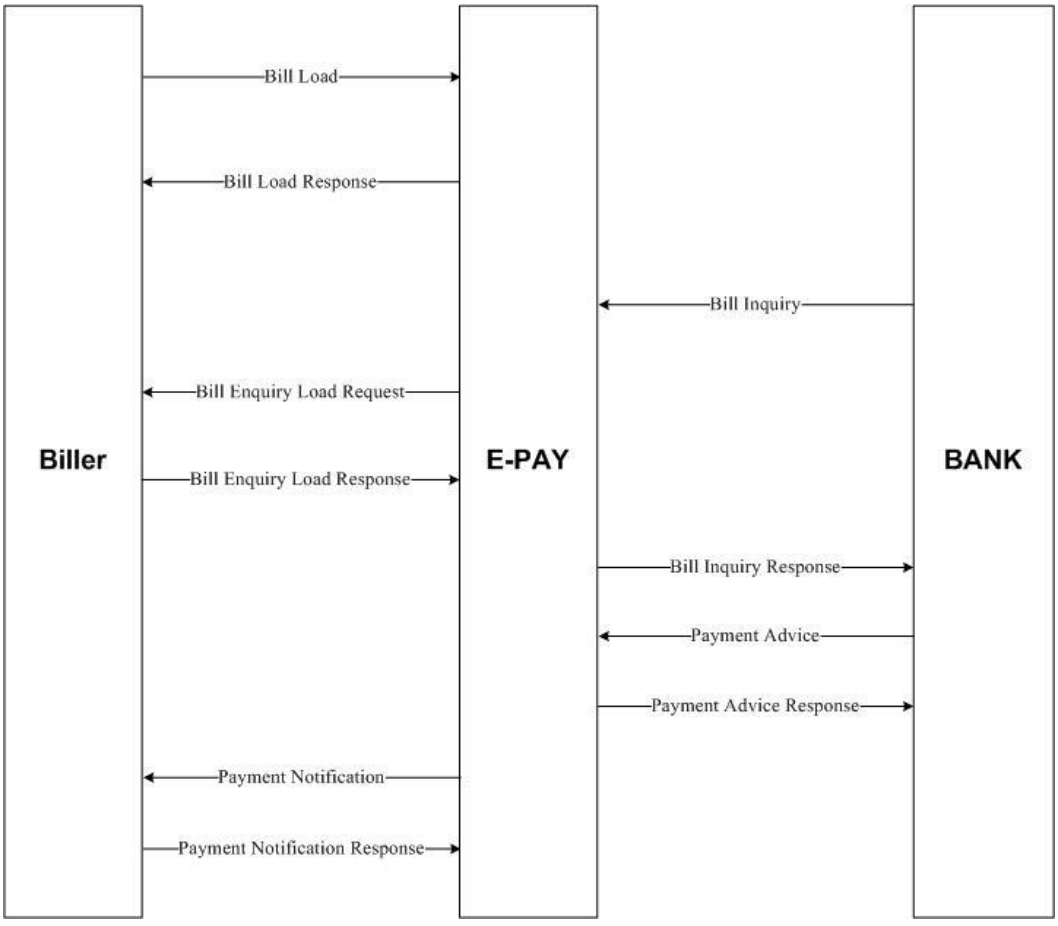
Bill Payment Service will compute monetary amounts due to each Biller/Bank. Settlements occur when Bill Payment Service sends fund transfer instructions to the Settlement Bank that debits the Banks, and credit the corresponding Billers.

2.2. Business Process Flow

In the following section, you can find a detailed description of the service process flow from the business point of view, and it is classified based on the mechanism of receiving the service.

2.2.1. Real Time Process Flow

The real time process flow describes how different parties can interact with the system in an online messaging way, below is a step by step process flow that describe the exchanged messages and its responses.



3. Exchanged Messages

The following is a list of messages that shall be exchanged with the payment system from the bank side, where the messages are exchanged based on the an XML based massage transferred using a Web Service.

3.1. Bank Interface Messages

Message	From	To
Real Time Mode	-	-
Bill Inquiry	-	-
Bill Inquiry (BillInqRq Message)	Bank	EPAY
Bill Inquiry Response (BillInqRs Message)	EPAY	Bank
Payment Advice	-	-
Payment Advice (PmtAdviceRq Message)	Bank	EPAY
Payment Advice Response (PmtAdviceRs Message)	EPAY	Bank

4. Data Types

Data Type	Description	Format
Boolean	Boolean indicates a logical True or False condition.	True False
Character	Character indicates an element that allows character data up to a maximum number of characters	For example, C-12
Narrow Character	Elements of type Narrow Character are elements of character data type with the additional restriction that the only allowable characters are those contained within the ISO Latin-1 character set	For example, NC-30
Currency	All currency amounts are required to have exactly two digits to the right of the decimal place	For example , 437.90
Date	YYYY-MM-DDTHH:MM:SS	2009-03-19T19:49:52
Decimal	The value is up to fifteen (15) decimal digits in length. The value is not restricted to integer values and has a decimal point The sign is always optional - if it is absent, the value is assumed to be positive.	+.12345678901234 +123456.734
Closed Enum	A Closed Enum is an element where a number of valid values are defined within the specification.	Max (20 Char)
Open Enum	An Open Enum is an element where a number of valid values are defined within this specification, but other values should not be rejected as invalid	Max (20 Char)

	by any system, but must be defined in the system before the first use, and an update list will be sent to service participants.	
Identifier	This is a Narrow Character data type with a maximum length of 36.	
Long	Long is an integer value, which may be positive, negative, or zero, with values ranging from +2147483647 to -2147483648.	
Phone	Phone Number indicates a string of up to 32 narrow characters in length (NC-32).	+CountryCode-CityCode-LocalPhoneNumber+PBXextension +966-03-8875111+212
UUID	UUID elements are Narrow Character with a maximum length of 36.	
URL	A URL is of the Narrow Character data type with a length of 1024 Characters (NC-1024)	The format of a URL begins with a string that identifies which protocol is to be used to access the information, such as “http://”.

5.

5. Bank Interface

The bill payment and presentment system enable banks to exchange files and online messages that represent two cycles, the presentment cycle to enquire for bills either from the bill store or directrly from the biller, based on the enquiry type and the biller mode.

The system supports the following enquiry types:

1. Customer specific enquiry: where the customer would like to collect all his bills from the bill store without selecting aspecific biller.
2. Biller specific enquiry: where the customer specify a biller to enquire for, in this case the bill will be retrieved based on the biller mode as follow:
    a. Offline Biller
    The biller use the bill store to upload his bill, and only bill existing in the bill store will be retrived.
    b. Online Biller
    The biller is online connected to the e-Pay system, no bills are stored in the bill store, and for any bank enquiry, the e-Pay system send an enquiry request to the biller to retrieve the bill.
    c. Online/Offline Biller
    The biller is online connected to the e-Pay system, while he is able to upload bills into the bill store, and for any bank enquiry, the e-Pay system check for the bill in the bill store, if bill not found, the e-Pay system send an enquiry request to the biller to retrieve the bill.

The other cycle is the payment cycle, where the bank confirms a payment completion from his side.

5.1. Real Time Mode

The real time mode enable the bank to exchnage messages with the e-Pay system, the messages are exchanged based on a SOAP based web service interface that carry on the below exchanged XML based messages.

5.1.1. Bill Inquiry

The bill enquiry cycle enable the bank to send an online bill enquiry to e-Pay system to enquire either for customer specific bills or for biller specific bills.

5.1.1.1. Bill Inquiry (BillInqRq Message)

The bill enquiry request is prepared by the bank and sent to e-pay system reflecting the bill reference parameters that the bank would liketo enquire for.

Element	Type	Cardinality
EFBPS	<EFBPS>	1
EFBPS/SignonRq	<SignonRq>	1
EFBPS/SignonRq/ClientDt	Date	1
EFBPS/SignonRq/LanguagePref	Open Enum	0..1
EFBPS/SignonRq/SignonProfile	<SignonProfile>	1
EFBPS/SignonRq/SignonProfile/Sender	Identifier	1
EFBPS/SignonRq/SignonProfile/Receiver	Identifier	0..1
EFBPS/SignonRq/SignonProfile/MsgCode	Identifier	1
EFBPS/BankSvcRq	<BankSvcRq>	1
EFBPS/BankSvcRq/RqUID	UUID	1
EFBPS/BankSvcRq/BillInqRq	<BillInqRq>	1
EFBPS/BankSvcRq/BillInqRq/BankId	NC-8	0..1
EFBPS/BankSvcRq/BillInqRq/BranchCode	NC-10	0..1
EFBPS/BankSvcRq/BillInqRq/DistrictCode	NC-10	0..1
EFBPS/BankSvcRq/BillInqRq/AccessChannel	Open Enum	1
EFBPS/BankSvcRq/BillInqRq/CustId	<CustId>	0..1
EFBPS/BankSvcRq/BillInqRq/CustId/OfficialId	C-32	1
EFBPS/BankSvcRq/BillInqRq/CustId/OfficialIdType	Open Enum	0..1
EFBPS/BankSvcRq/BillInqRq/ProxyCustId	<CustId>	0..1
EFBPS/BankSvcRq/BillInqRq/ProxyCustId/OfficialId	C-32	1
EFBPS/BankSvcRq/BillInqRq/ProxyCustId/OfficialIdType	Open Enum	0..1
EFBPS/BankSvcRq/BillInqRq/BillNumber	C-32	0..1
EFBPS/BankSvcRq/BillInqRq/AccountId	<AccountId>	0..1
EFBPS/BankSvcRq/BillInqRq/AccountId/BillingAcct	C-32	0..1
EFBPS/BankSvcRq/BillInqRq/AccountId/BillerId	NC-4	1
EFBPS/BankSvcRq/BillInqRq/ServiceType	Open Enum	0..1
EFBPS/BankSvcRq/BillInqRq/DateRange	<DateRange>	0..1
EFBPS/BankSvcRq/BillInqRq/DateRange/StartDt	Date	0..1
EFBPS/BankSvcRq/BillInqRq/DateRange/EndDt	Date	0..1
EFBPS/BankSvcRq/BillInqRq/IncPayments - FALSE by default	Boolean	0..1
EFBPS/BankSvcRq/BillInqRq/IncPaidBills - FALSE by default	Boolean	0..1
EFBPS/BankSvcRq/BillInqRq/IncBillSummAmt - FALSE by default	Boolean	0..1
EFBPS/BankSvcRq/BillInqRq/IncExactPmt - FALSE by default	Boolean	0..1
EFBPS/BankSvcRq/BillInqRq/MaxBills	Long	0..1
EFBPS/BankSvcRq/BillInqRq/AdditionInfo	<AdditionInfoType>	0..1
EFBPS/BankSvcRq/BillInqRq/AdditionInfo/Values	<ValuesType>	0..2
EFBPS/BankSvcRq/BillInqRq/AdditionInfo/Values/ Key	Char255Type	0..1
EFBPS/BankSvcRq/BillInqRq/AdditionInfo/Values/ Value	String	0..1

- **MsgCode**

Name	Value
MessageCode	RBINQRQ

- **Customer and Proxy User Data**

Bill queries actually contain two distinct CustId aggregates: CustId and ProxyCustId. The CustId aggregate identifies the bill-paying Customer, whereas the ProxyCustId represents an individual on behalf of that Customer. Examples of a proxy Customer include Bank Tellers, and other similar individuals. Refer to the OfficialIdType enum for a list of valid proxy Customer types.

The CustId aggregate may be NULL for bill specific queries, or optionally provided. The ProxyCustId is always NULL unless the query is performed by a third party. Bank applications must enforce the use of this aggregate so that Bank employee usage can be properly monitored.

- **Customer / Bill Specific Queries**

Customer-specific bill queries are implied by the absence of specific bill data and inclusion of Customer identification in the bill query. The absence of both data points is invalid, but the inclusion of both results in a bill specific query.

A Bill-specific query requires a specific account number from which bill data is to be fetched. An account number is typically sufficient to identify the desired bill, but an additional bill number is sometimes required as some Billers associate several active bills of the same category with a single account.

Bills are always fetched from the most recent bill cycle (if applicable) unless a date range is provided in the query. As such, it is possible to perform queries on historical bill and payment data.

- **Include Payment History**

Banks may determine whether payment history is to be included in the query response, or perhaps leave this option available for the Customer to choose. Payment history is not displayed by default, so this option must be explicitly included for such data to be returned.

- **Include Paid Bills**

The IncPaidBills value permits inclusion of deactivated bills and Customer Initiated Payments. This value has no effect on bill specific queries as the specified bills in such scenarios will always be displayed (for example, historical bills are fetched using date range parameters). The IncPaidBills is assumed FALSE if not explicitly included in the query.

- **Include Bill Summary Data**

The BillSummAmt aggregate contains high-level bill specific detail which may be relevant to the access channel and/or Customer. EPAY may be instructed not to convey this information, but it is not included by default.

- **Include Exact Payment**

This flag permits the sending application to have determine, at the time of query, if exact payment amount is required. This task is typically performed at time of payment confirmation.

- **MaxBills**

The MaxBills value permits the Bank application to limit the number of bills which may be retrieved for any specific query.

5.1.1.2. Bill Inquiry Response (BillInqRs Message)

The e-Pay system respond to the bank bill enquiry request using the following message, if the search results in found bills, the response will contains a list of the bills that match the search criteria.

Element	Type	Cardinality
EFBPS	<EFBPS>	1
EFBPS/SignonRq	<SignonRq>	1
EFBPS/SignonRq/ClientDt	Date	0..1
EFBPS/SignonRq/LanguagePref	Open Enum	0..1
EFBPS/SignonRq/SignonProfile	<SignonProfile>	1
EFBPS/SignonRq/SignonProfile/Sender	Identifier	1
EFBPS/SignonRq/SignonProfile/Receiver	Identifier	0..1
EFBPS/SignonRq/SignonProfile/MsgCode	Identifier	1
EFBPS/BankSvcRs	<BankSvcRs>	1
EFBPS/BankSvcRs/Status	Status	1
EFBPS/BankSvcRs/Status/StatusCode	Long	1
EFBPS/BankSvcRs/Status/ShortDesc	C-255	0..1
EFBPS/BankSvcRs/RqUID	UUID	1
EFBPS/BankSvcRs/BillInqRs	<BillInqRs>	1
EFBPS/BankSvcRs/BillInqRs/Msg	<Msg>	0..1
EFBPS/BankSvcRs/BillInqRs/Msg/LanguagePref	Language	1
EFBPS/BankSvcRs/BillInqRs/Msg/Text	C-4096	1
EFBPS/BankSvcRs/BillInqRs/RecCount	Long	1
EFBPS/BankSvcRs/BillInqRs/BillRec	<BillRec>	0..*
EFBPS/BankSvcRs/BillInqRs/BillRec/SupercedeBillNumber	C-32	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillPresentmentStatusCode	Closed Enum	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillPmtStatus	<BillPmtStatus>	0..*
EFBPS/BankSvcRs/BillInqRs/BillRec/BillPmtStatus/PmtTransId	<PmtTransId>	1..3
EFBPS/BankSvcRs/BillInqRs/BillRec/BillPmtStatus/PmtTransId/PmtId	NC-32	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillPmtStatus/PmtTransId/PmtIdType	Closed Enum	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillPmtStatus/PmtStatusCode	Open Enum	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillPmtStatus/PayAmt	<CurAmt>	1..*
EFBPS/BankSvcRs/BillInqRs/BillRec/BillPmtStatus/PayAmt/Sequence	NC-2	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillPmtStatus/PayAmt/Amt	Currency	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillPmtStatus/PayAmt/CurCode	NC-3	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillPmtStatus/EffDt	Date	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/CustId	<CustId>	0..*
EFBPS/BankSvcRs/BillInqRs/BillRec/CustId/OfficialId	C-32	1
EFBPS/BankSvcRs/BillInqRs/BillRec/CustId/OfficialIdType	Open Enum	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/DisplayLabel	<DisplayLabel>	0..*
EFBPS/BankSvcRs/BillInqRs/BillRec/DisplayLabel\LanguagePref	Open Enum	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/DisplayLabel\Text	C-32	1
EFBPS/BankSvcRs/BillInqRs/BillRec/Msg	<Msg>	0..*
EFBPS/BankSvcRs/BillInqRs/BillRec/Msg/LanguagePref	Language	1
EFBPS/BankSvcRs/BillInqRs/BillRec/EPayBillRecID	UUID	1
EFBPS/BankSvcRs/BillInqRs/BillRec/Msg/Text	C-4096	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo	<BillInfo>	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/BillCategory	Open Enum	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/ServiceType	Open Enum	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/BillCycle	NC-16	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/BillNumber	C-32	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/AccountId	<AccountId>	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/AccountId/BillingAcct	C-32	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/AccountId/BillerId	NC-4	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt (AmountDue)	<CurAmt>	1..*
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/Sequence	NC-2	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/OriginalAmt	Currency	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/AmtDue	Currency	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/CurCode	NC-3	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/ExactPmt	Boolean	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/PymtMode	Closed Enum	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/RangeInfo	< RangeInfo>	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/RangeInfo/MinAmt	Currency	1

EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/RangeInfo/MaxAmt	Currency	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/InstInfo	< InstInfo>	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/InstInfo/NumInst	NC-2	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/InstInfo/InstAmt	Currency	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/InstInfo/MinInst	NC-2	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/ShortDesc	<ShortDesc>	0..*
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/ShortDesc/LanguagePref	Language	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CurAmt/ShortDesc/Text	C-255	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/DueDt	Date	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/OpenDt	Date	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/CloseDt	Date	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/ExpDt	Date	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/BillRefInfo	NC-80	0..1
EFBPS/BankSvcRs/BulkBillInqRs/BillRec/BillInfo/BillSummAmtGroup	<BillSummAmtGroup>	0..*
EFBPS/BankSvcRs/BulkBillInqRs/BillRec/BillInfo/BillSummAmtGroup/GroupSeq	NC-2	1
EFBPS/BankSvcRs/BulkBillInqRs/BillRec/BillInfo/BillSummAmtGroup/ShortDesc	<ShortDesc>	1..*
EFBPS/BankSvcRs/BulkBillInqRs/BillRec/BillInfo/BillSummAmtGroup/ShortDesc/ LanguagePref	Language	1
EFBPS/BankSvcRs/BulkBillInqRs/BillRec/BillInfo/BillSummAmtGroup/ShortDesc/ Text	C-255	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/BillSummAmtGroup/BillSummAmt	<BillSummAmt>	1..*
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/BillSummAmtGroup/BillSummAmt/Amt	Currency	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/BillSummAmtGroup/BillSummAmt/ CurCode	NC-3	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/BillSummAmtGroup/BillSummAmt/ CurRate	Currency	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/BillSummAmtGroup/BillSummAmt/ BillSummAmtCode	Open Enum	0..1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/BillSummAmtGroup/BillSummAmt/ BillSummAmtType	Open Enum	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/BillSummAmtGroup/BillSummAmt/ ShortDesc	<ShortDesc>	0..*
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/BillSummAmtGroup/BillSummAmt/ ShortDesc/LanguagePref	Language	1
EFBPS/BankSvcRs/BillInqRs/BillRec/BillInfo/BillSummAmtGroup/BillSummAmt/ ShortDesc/Text	C-255	1
EFBPS/BankSvcRs/BillInqRs/AdditionInfo	<AdditionInfoType>	0..1
EFBPS/BankSvcRs/BillInqRs/AdditionInfo/Values	<ValuesType>	0..2
EFBPS/BankSvcRs/BillInqRs/AdditionInfo/Values/ Key	Char255Type	0..1
EFBPS/BankSvcRs/BillInqRs/AdditionInfo/Values/ Value	String	0..1

- **MsgCode**

Name	Value
MessageCode	RBINQRS

- **Bill Messages**

EPAY permits two types of messages to be associated with each BillInqRs message. The first is the Global message which resides directly below the BillInqRs tag. Global messages apply to the entire query results and are defined exclusively by EPAY.

Messages may also be associated with individual bills. Such messages are typically provided by the Biller and are limited to just one message per bill.

- **EPayBillRecID**

The EPayBillRecID represents a unique identifier for the bill, and will be used in all subsequent messages to confirm the bill payment.

- **BillPmtStatus**

The BillPmtStatus aggregate is used to convey payment history on a per-bill basis. A minimum of two Payment Transaction Numbers will be included with each payment. One such number will be derived from either the Bank or Biller applications, but the other will always be the EPAY Payment Transaction Number as every payment in the EPAY system is assigned such a number.

- **BillPresentmentStatusCode**

The BillPresentmentStatusCode enum is used differently in bill upload, presentment, and other scenarios. For bill queries, the aggregate informs the Customer whether the bill has been partially paid, fully paid, or not paid at all. Refer to the BillStatusCode enum for possible values contained within the aggregate.

- **DisplayLabel**

Only a single language version of a bill's display label is provided in this message. The language provided is that which was requested in the BillInqRq message. This language is also provided in BillInqRs/SignonRs/LanguagePref element.

- **BillInqRs Status Values**

Code	Description
0	Request processed successfully
1	Request processed successfully, but fetch limit reached. Some records may not have been fetched
8001	XML. Schema Validation Failed
8002	Invalid Sender Code
8003	Invalid Receiver Code
8004	Invalid Message Code
8005	Repeated RqUID
8006	Invalid Bank Id
8007	Invalid Branch Code
8008	Invalid Access Channel Code
8009	Invalid Customer Id's Official ID Type
8010	Invalid Proxy Customer Id's Official ID
8011	Biller ID is Mandatory, When Providing the Bill Number
8012	Invalid Biller Id
8013	Neither Billing Account nor Customer ID Specified In The Query
8014	Invalid Service Type Or Not Mapped To Biller
8015	Start Date is Greater Than The End Date
8016	Either Start or End date Is Mandatory In Case of DateRange Element Is Available
8017	The Max Number of Bills Limitation Reached
8018	No Valid Fulfillment Entity Settlement Account To Be Used For Payment Settlement
8019	Sender Not Active
8020	Bank Not Active
8090	Biller Enquiry service temporary not available (mapped to online enquiry reason code 2097)
8091	Biller not support enquiry by customer id (mapped to online enquiry reason code 2007)
8092	Biller not support enquiry by account number (mapped to online enquiry reason code 2006)
8093	Biller not support enquiry by bill number (mapped to online enquiry reason code 2005)
8094	Bill Exists, but can't be paid (mapped to online enquiry reason code 2004)
8095	Invalid Response Received From Online Biller, or Biller Unable to Process The Request
8096	No Response Received From Biller Online Enquiry, Try Later.
8097	No Matched Customer Found

	(mapped to online enquiry reason code 2002)
8098	No Bill Found For The Mentioned Customer (mapped to online enquiry reason code 2003)
8099	Bill Not Found (if no match found in bill store, or mapped to online enquiry reason code 2001)

5.1.2. Payment Advice

The bank sends the payment advice message to the e-Pay system to confirm a bill payment completion to be recorded in the system, and to be considered for the current business day settlement.

5.1.2.1. Payment Advice (PmtAdviceRq Message)

The payment advice message represents the bank payment confirmation to the e-pay system, where the e-Pay system will try to match this request with an existing bill store bill.

Element	Type	Cardinality
EFBPS	<EFBPS>	1
EFBPS/SignonRq	<SignonRq>	1
EFBPS/SignonRq/ClientDt	Date	1
EFBPS/SignonRq/LanguagePref	Open Enum	0..1
EFBPS/SignonRq/SignonProfile	<SignonProfile>	1
EFBPS/SignonRq/SignonProfile/Sender	Identifier	1
EFBPS/SignonRq/SignonProfile/Receiver	Identifier	0..1
EFBPS/SignonRq/SignonProfile/MsgCode	Identifier	1
EFBPS/BankSvcRq	<BankSvcRq>	1
EFBPS/BankSvcRq/RqUID	UUID	1
EFBPS/BankSvcRq/PmtAdviceRq	<PmtAdviceRq>	1
EFBPS/BankSvcRq/PmtAdviceRq/ProxyCustId	<CustId>	0..1
EFBPS/BankSvcRq/PmtAdviceRq/ProxyCustId/OfficialId	C-32	1
EFBPS/BankSvcRq/PmtAdviceRq/ProxyCustId/OfficialIdType	Open Enum	0..1
EFBPS/BankSvcRq/PmtAdviceRq/EPayBillRecID	UUID	1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec	<PmtRec>	1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtTransId	<PmtTransId>	1..3
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtTransId/PmtId	NC-32	1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtTransId/PmtIdType	Closed Enum	0..1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/CustId	<CustId>	0..1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/CustId/OfficialId	C-32	1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/CustId/OfficialIdType	Open Enum	0..1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo	<PmtInfo>	1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/PayAmt	<CurAmt>	1..*
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/PayAmt/Sequence	NC-2	1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/PayAmt/Amt	Currency	1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/PayAmt/CurCode	NC-3	1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/FeesAmt	<PayAmt>	0..*
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/FeesAmt/Amt	Currency	1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/FeesAmt/CurCode	NC-3	1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/PrcDt	Date	1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/BillCycle	NC-16	0..1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/BillNumber	C-32	0..1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/AccountId	<AccountId>	1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/AccountId/BillingAcct	C-32	1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/AccountId/BillerId	NC-4	0..1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/BankId	NC-8	0..1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/DistrictCode	NC-10	0..1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/BranchCode	NC-10	0..1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/AccessChannel	Open Enum	0..1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/PmtMethod	Open Enum	0..1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/ServiceType	Open Enum	0..1
EFBPS/BankSvcRq/PmtAdviceRq/PmtRec/PmtInfo/PmtRefInfo	NC-80	0..1
EFBPS/BillerSvcRq/PmtNotifyRq/PmtRec/PmtInfo/AdditionInfo	<AdditionInfoType>	0..1
EFBPS/BillerSvcRq/PmtNotifyRq/PmtRec/PmtInfo/AdditionInfo/Values	<ValuesType>	
EFBPS/BillerSvcRq/PmtNotifyRq/PmtRec/PmtInfo/AdditionInfo/Values/ Key	Char255Type	
EFBPS/BillerSvcRq/PmtNotifyRq/PmtRec/PmtInfo/AdditionInfo/Values/ Value	String	

- **MsgCode**

Name	Value
MessageCode	RPADVRQ

- **PmtRec**

The PmtRec aggregate is simply a wrapper for the PmtInfo aggregate and is used to add values which reflect the fact the payment has actually been accepted. For example, the enveloping PmtRec message contains a PmtTransId aggregate and must contain both the EPAY Payment Transaction Number and Bank Transaction Number for the payment.

- **PmtRec**

The PmtRec aggregate is simply a wrapper for the PmtInfo aggregate and is used to add values which reflect the fact the payment has actually been accepted. For example, the enveloping PmtRec message contains a PmtTransId aggregate and must contain both the EPAY Payment Transaction Number and Bank Transaction Number for the payment.

- **Customer and Proxy User Data**

Bill queries actually contain two distinct CustId aggregates: CustId and ProxyCustId. The CustId aggregate identifies the bill-paying Customer, whereas the ProxyCustId represents an individual on behalf of that Customer. Examples of a proxy Customer include Bank Tellers, and other similar individuals. Refer to the OfficialIdType enum for a list of valid proxy Customer types.

The CustId aggregate may be NULL for bill specific queries, or optionally provided. The ProxyCustId is always NULL unless the query is performed by a third party. Bank applications must enforce the use of this aggregate so that Bank employee usage can be properly monitored.

- **PmtTransId**

The Bank is required to provide a Payment Transaction Numbers for each payment in the advice message.

5.1.2.2. Payment Advice Response (PmtAdviceRs Message)

The payment advice response represents a response for a previously uploaded payment advice, and reflects the processing status of the sent request; the payment will be accepted only if the bill status is still valid for payment.

Element	Type	Cardinality
EFBPS	<EFBPS>	1
EFBPS/SignonRq	<SignonRq>	1
EFBPS/SignonRq/ClientDt	Date	0..1
EFBPS/SignonRq/LanguagePref	Open Enum	0..1
EFBPS/SignonRq/SignonProfile	<SignonProfile>	1
EFBPS/SignonRq/SignonProfile/Sender	Identifier	1
EFBPS/SignonRq/SignonProfile/Receiver	Identifier	0..1
EFBPS/SignonRq/SignonProfile/MsgCode	Identifier	1
EFBPS/BankSvcRs	<BankSvcRs>	1
EFBPS/BankSvcRs/RqUID	UUID	1
EFBPS/BankSvcRs/PmtAdviceRs	<PmtAdviceRs>	1
EFBPS/BankSvcRs/PmtAdviceRs/PmtRecAdviceStatus	<PmtRecAdviceStatus>	1
EFBPS/BankSvcRs/PmtAdviceRs/PmtRecAdviceStatus/PmtTransId	<PmtTransId>	1..2
EFBPS/BankSvcRs/PmtAdviceRs/PmtRecAdviceStatus/PmtTransId/PmtId	NC-32	1
EFBPS/BankSvcRs/PmtAdviceRs/PmtRecAdviceStatus/PmtTransId/PmtIdType	Closed Enum	0..1
EFBPS/BankSvcRs/PmtAdviceRs/PmtRecAdviceStatus/Status	<Status>	1
EFBPS/BankSvcRs/PmtAdviceRs/PmtRecAdviceStatus/Status/StatusCode	Long	1
EFBPS/BankSvcRs/PmtAdviceRs/PmtRecAdviceStatus/Status/ShortDesc	C-255	0..1
EFBPS/BankSvcRs/PmtAdviceRs/PmtRecAdviceStatus/SettlDt	Date	0..1
EFBPS/BankSvcRs/ PmtAdviceRs /AdditionInfo	<AdditionInfoType>	0..1
EFBPS/BankSvcRs/ PmtAdviceRs /AdditionInfo/Values	<ValuesType>	0..2
EFBPS/BankSvcRs/ PmtAdviceRs /AdditionInfo/Values/ Key	Char255Type	0..1
EFBPS/BankSvcRs/ PmtAdviceRs /AdditionInfo/Values/ Value	String	0..1

- **MsgCode**

Name	Value
MessageCode	RPADVRS

- **Status**

There are two distinct usages of the Status aggregate in the PmtAdviceRs message. The first pertains to the overall message and provides indication of any type of problem in processing the PmtAdviceRq message.

The Status aggregate within the PmtRecAdviceStatus aggregate, however, provides feedback on the processing of individual payments within the corresponding PmtAdviceRq message. The unsuccessful processing of any single payment in the PmtAdviceRq message will cause all payments in the advice message to fail.

- **PmtAdviceRs Status Codes**

Code	Description
0	Payment Advice Accepted
9001	Invalid Sender Code
9002	Invalid Receiver Code
9003	Invalid Message Code
9004	Repeated RqUID
9005	Invalid Proxy Customer ID's Official ID Type
9006	Unable to Match Payment With Existing Bill
9007	Repeated PmtId
9008	Invalid Cust Id's Official Id Type

9009	PayAmt's Sequence Not Matched With An Existing Bill's Currency Amount
9010	Missing Exact Payment Currency Amount
9011	PayAmt Can't Be Zero
9012	For Range Payment Mode, Payment Amount Should Be Greater Than or Equal To The Minimum Amount
9013	For Range Payment Mode, and Maximum Amount Not Equal To Zero, Payment Amount Should Be Less Than or Equal Maximum Amount
9014	For Installment Payment Mode, Payment Amount Should Be Multiple of The Installment Amount or Equal To The Amount Due
9015	PayAmt Curcode Not Matched With Original Amount Due Master
9016	Not Valid Fees Amount
9017	Not Valid Fees Currency
9018	Invalid PrcDt, It Must be Grater Than or Equal The Current Business Date
9019	Invalid Bank Id
9020	Invalid Branch Code
9021	Invalid Access Channel Code
9022	Invalid PmtMethod Code
9023	Invalid Check Digit Value
9024	Invalid Service Type Code
9025	Invalid PmtRefInfo
9026	No Valid Bank Settlement Accounts
9027	Bill Status Does Not Allow Payment
9028	Bill In Hold State, Waiting For Biller Update
9029	Bill Had Expired, or Not Yet Valid For Payment
9030	The payments legnth is not valid
9031	Repeated Sequence
9032	Invalid Pmt Id Type Code
9033	Invalid Bill Number
9034	Invalid Billing Account
9035	Invalid Biller Id
9036	Invalid Message Code
9037	Bill already Paid
9038	Fractions is not allowed
9039	Invalid Payment Sequence
9040	The Customer Payment Name Is Required

5.1.3. EPay Fees Inquiry

The fees inquiry cycle enable the bank to send an online fee enquiry to e-Pay system to calculate the e-Pay commission for the bank's customer selected biller and based on the final payment amounts selected to be paid.

5.1.3.1. Fee Inquiry Request (FeeInqRq Message)

Element	Type	Cardinality
EFBPS	<EFBPS>	1
EFBPS/SignonRq	<SignonRq>	1
EFBPS/SignonRq/ClientDt	Date	1
EFBPS/SignonRq/LanguagePref	Open Enum	0..1
EFBPS/SignonRq/SignonProfile	<SignonProfile>	1
EFBPS/SignonRq/SignonProfile/Sender	Identifier	1
EFBPS/SignonRq/SignonProfile/Receiver	Identifier	0..1
EFBPS/SignonRq/SignonProfile/MsgCode	Identifier	1
EFBPS/BankSvcRq	<BankSvcRq>	1
EFBPS/BankSvcRq/RqUID	UUID	1
EFBPS/BankSvcRq/FeeInqRq	<FeeInqRq>	1
EFBPS/BankSvcRq/FeeInqRq/EPayBillRecID	UUID	1
EFBPS/BankSvcRq/FeeInqRq/PayAmt	<PayAmt>	1..*
EFBPS/BankSvcRq/FeeInqRq/PayAmt/Sequence	NC-2	1
EFBPS/BankSvcRq/FeeInqRq/PayAmt/Amt	Currency	1
EFBPS/BankSvcRq/FeeInqRq/PayAmt/CurCode	NC-3	1
EFBPS/BankSvcRq/FeeInqRq/AdditionInfo	<AdditionInfoType>	0..1

EFBPS/BankSvcRq/FeeInqRq/AdditionInfo/Values	<ValuesType>	0..2
EFBPS/BankSvcRq/FeeInqRq/AdditionInfo/Values/ Key	Char255Type	0..1
EFBPS/BankSvcRq/FeeInqRq/AdditionInfo/Values/ Value	String	0..1

- **MsgCode**

Name	Value
MessageCode	RFINQRQ

- **EPayBillRecID**

The Bank is required to provide the EPayBillRecID "for the bill selected to be paid", that has been provided by the BillInqRs Message.

- **PayAmt**

The list of PayAmt aggregate represents the amounts that have been selected and confirmed by the bank's customer to be paid, and will be used for calculating e-Pay commission.

5.1.3.2. Fee Inquiry Response (FeeInqRs Message)

Element	Type	Cardinality
EFBPS	<EFBPS>	1
EFBPS/SignonRq	<SignonRq>	1
EFBPS/SignonRq/ClientDt	Date	1
EFBPS/SignonRq/LanguagePref	Open Enum	0..1
EFBPS/SignonRq/SignonProfile	<SignonProfile>	1
EFBPS/SignonRq/SignonProfile/Sender	Identifier	1
EFBPS/SignonRq/SignonProfile/Receiver	Identifier	0..1
EFBPS/SignonRq/SignonProfile/MsgCode	Identifier	1
EFBPS/BankSvcRs	<BankSvcRs>	1
EFBPS/BankSvcRs/Status	Status	1
EFBPS/BankSvcRs/Status/StatusCode	Long	1
EFBPS/BankSvcRs/Status/ShortDesc	C-255	0..1
EFBPS/BankSvcRs/RqUID	UUID	1
EFBPS/BankSvcRs/FeeInqRs	<FeeInqRs>	0..1
EFBPS/ BankSvcRs /FeeInqRs/EPayBillRecID	UUID	1
EFBPS/ BankSvcRs /FeeInqRs/FeesAmt	<FeesAmt>	0..*
EFBPS/ BankSvcRs /FeeInqRs/FeesAmt/Amt	Currency	1
EFBPS/ BankSvcRs /FeeInqRs/FeesAmt/CurCode	NC-3	1
EFBPS/ BankSvcRs / FeeInqRs /AdditionInfo	<AdditionInfoType>	0..1
EFBPS/ BankSvcRs / FeeInqRs /AdditionInfo/Values	<ValuesType>	0..2
EFBPS/ BankSvcRs / FeeInqRs /AdditionInfo/Values/ Key	Char255Type	0..1
EFBPS/ BankSvcRs / FeeInqRs /AdditionInfo/Values/ Value	String	0..1

- **MsgCode**

Name	Value
MessageCode	RFINQRS

- **EPayFeeAmount**

The list of FeesAmt aggregate represents the commission required for the e-Pay system, and grouped by fees settlement currency, for empty list of FeesAmt, e-Pay commission is zero.

- **FeeInqRs Status Values**

Code	Description
0	Request processed successfully
1	Request processed successfully, but fetch limit reached. Some records may not have been fetched
1102	Invalid Sender Code
1103	Invalid Receiver Code
1104	Invalid Message Code
1105	Sender Not Active
1106	Repeated RqUID
1107	Invalid EPayBillRecID
1108	Bill Not Found
1109	Bill Can Not Be Paid
1110	Invalid Payment Amounts
1111	No Commission Configured (for all or some of the payment currencies)
1112	The payments legnth is not valid
1113	Repeated Sequence
1114	PayAmt Curcode Not Matched With Original Amount Due Master
1115	For Range Payment Mode, Payment Amount Should Be Greater Than or Equal To The Minimum Amount
1116	For Range Payment Mode, and Maximum Amount Not Equal To Zero, Payment Amount Should Be Less Than or Equal Maximum Amount
1117	For Installment Payment Mode, Payment Amount Should Be Multiple of The Installment Amount or Equal To The Amount Due
1118	Missing Exact Payment Currency Amount
1119	PayAmt's Sequence Not Matched With An Existing Bill's Currency Amount
1120	Invalid Bank Id

1121	Fractions is not allowed
1122	Invalid Payment Sequence
1199	General Error

5.1.4. All Billers Inquiry

All Billers inquiry allows banks to retrieve full Billers Heirarchy starting from Service Groups, Service Types then Billers.

5.1.4.1. Get Billers Request (BillersInqRq message)

Element	Type	Cardinality
EFBPS	<EFBPS>	1
EFBPS/SignonRq	<SignonRq>	1
EFBPS/SignonRq/ClientDt	Date	1
EFBPS/SignonRq/LanguagePref	Open Enum	0..1
EFBPS/SignonRq/SignonProfile	<SignonProfile>	1
EFBPS/SignonRq/SignonProfile/Sender	Identifier	1
EFBPS/SignonRq/SignonProfile/Receiver	Identifier	0..1
EFBPS/SignonRq/SignonProfile/MsgCode	Identifier	1
EFBPS/BankSvcRq	<BankSvcRq>	1
EFBPS/BankSvcRq/RqUID	UUID	1

- **MsgCode**

Name	Value
MessageCode	BRSINQRQ

5.1.4.2. Get Billers Response (BillersInqRs Message)

Element	Type	Cardinality
EFBPS	<EFBPS>	1
EFBPS/SignonRq	<SignonRq>	1
EFBPS/SignonRq/ClientDt	Date	1
EFBPS/SignonRq/LanguagePref	Open Enum	0..1
EFBPS/SignonRq/SignonProfile	<SignonProfile>	1
EFBPS/SignonRq/SignonProfile/Sender	Identifier	1
EFBPS/SignonRq/SignonProfile/Receiver	Identifier	0..1
EFBPS/SignonRq/SignonProfile/MsgCode	Identifier	1
EFBPS/BankSvcRs	<BankSvcRs>	1
EFBPS/BankSvcRs/Status	Status	1
EFBPS/BankSvcRs/Status/StatusCode	Long	1
EFBPS/BankSvcRs/Status/ShortDesc	C-255	0..1
EFBPS/BankSvcRs/RqUID	UUID	1
EFBPS/BankSvcRs/BillersInqRs	<GetBillersRs>	0..1
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup	<ServiceGroup>	0..*
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup/Code	Identifier	1
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup/Name	Identifier	1
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup/ServiceTypeList	<ServiceTypeList>	0..1
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup/ServiceTypeList/ServiceType	<ServiceType>	0..*
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup/ServiceTypeList/ServiceType/Code	Identifier	1
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup/ServiceTypeList/ServiceType /Ar_Name	Identifier	1
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup/ServiceTypeList/ServiceType /En_Name	Identifier	1
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup/ServiceTypeList/ServiceType /BillerList	<BillerList>	0..1
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup/ServiceTypeList/ServiceType /BillerList/Biller	<Biller>	0..*
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup/ServiceTypeList/ServiceType /BillerList/Biller/Code	Identifier	1
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup/ServiceTypeList/ServiceType /BillerList/Biller/ParentBillerCode	Identifier	1
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup/ServiceTypeList/ServiceType /BillerList/Biller/Ar_Name	Identifier	1
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup/ServiceTypeList/ServiceType /BillerList/Biller/En_Name	Identifier	1
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup/ServiceTypeList/ServiceType /BillerList/Biller/SubBillerList	<SubBillerList>	0..1
EFBPS/BankSvcRs/BillersInqRs/ServiceGroup/ServiceTypeList/ServiceType /BillerList/Biller/SubBillerList/Biller	<Biller>	0..*

- **MsgCode**

Name	Value
MessageCode	BRSINQRS

- **GetBillersRs Status Values**

Code	Description
0	Request processed successfully
1	Request processed successfully, but fetch limit reached. Some records may not have been fetched
1102	Invalid Sender Code
1103	Invalid Receiver Code
1104	Invalid Message Code
1105	Sender Not Active
1106	Repeated RqUID
1199	General Error

5.1.5. Payment Inquiry Status

Payment Inquiry Status allows banks to retrieve payment confirmation response in case timeout occurs.

5.1.5.1. Payment Inquiry Request (PmtInqRq message)

Element	Type	Cardinality
EFBPS	<EFBPS>	1
EFBPS/SignonRq	<SignonRq>	1
EFBPS/SignonRq/ClientDt	Date	1
EFBPS/SignonRq/LanguagePref	Open Enum	0..1
EFBPS/SignonRq/SignonProfile	<SignonProfile>	1
EFBPS/SignonRq/SignonProfile/Sender	Identifier	1
EFBPS/SignonRq/SignonProfile/Receiver	Identifier	0..1
EFBPS/SignonRq/SignonProfile/MsgCode	Identifier	1
EFBPS/BankSvcRq	<BankSvcRq>	1
EFBPS/BankSvcRq/RqUID	UUID	1
EFBPS/BankSvcRq/PmtInqRq	< PmtInqRq >	1
EFBPS/BankSvcRq/PmtInqRq/BillerId	NC-5	1
EFBPS/BankSvcRq/PmtInqRq/EPayBillRecID	UUID	0..1
EFBPS/BankSvcRq/PmtInqRq/BillingAcct	C-32	0..1
EFBPS/BankSvcRq/PmtInqRq/BillNumber	C-32	0..1

- **MsgCode**

Name	Value
MessageCode	PMTINQRQ

5.1.5.2. Payment Inquiry Response (PmtInqRs message)

Element	Type	Cardinality
EFBPS	<EFBPS>	1
EFBPS/SignonRq	<SignonRq>	1
EFBPS/SignonRq/ClientDt	Date	0..1
EFBPS/SignonRq/LanguagePref	Open Enum	0..1
EFBPS/SignonRq/SignonProfile	<SignonProfile>	1
EFBPS/SignonRq/SignonProfile/Sender	Identifier	1
EFBPS/SignonRq/SignonProfile/Receiver	Identifier	0..1
EFBPS/SignonRq/SignonProfile/MsgCode	Identifier	1
EFBPS/BankSvcRs	<BankSvcRs>	1
EFBPS/BankSvcRs/RqUID	UUID	1
EFBPS/BankSvcRs/PmtInqRs/PmtAdviceRs	<PmtAdviceRs>	1
EFBPS/BankSvcRs/PmtInqRs/PmtAdviceRs/PmtRecAdviceStatus	<PmtRecAdviceStatus>	1
EFBPS/BankSvcRs/PmtInqRs/PmtAdviceRs/PmtRecAdviceStatus/PmtTransId	<PmtTransId>	1..2
EFBPS/BankSvcRs/PmtInqRs/ PmtAdviceRs/PmtRecAdviceStatus/PmtTransId/PmtId	NC-32	1
EFBPS/BankSvcRs/PmtInqRs PmtAdviceRs/PmtRecAdviceStatus/PmtTransId/PmtIdType	Closed Enum	0..1
EFBPS/BankSvcRs/PmtInqRs/PmtAdviceRs/PmtRecAdviceStatus/Status	<Status>	1
EFBPS/BankSvcRs/PmtInqRs PmtAdviceRs/PmtRecAdviceStatus/Status/StatusCode	Long	1
EFBPS/BankSvcRs/PmtInqRs PmtAdviceRs/PmtRecAdviceStatus/Status/ShortDesc	C-255	0..1
EFBPS/BankSvcRs/PmtInqRs/PmtAdviceRs/PmtRecAdviceStatus/SettlDt	Date	0..1
EFBPS/BankSvcRs/PmtInqRs/PmtAdviceRs/AdditionInfo	<AdditionInfoType>	0..1
EFBPS/BankSvcRs/PmtInqRs/PmtAdviceRs/AdditionInfo/Values	<ValuesType>	0..2
EFBPS/BankSvcRs/PmtInqRs/PmtAdviceRs/AdditionInfo/Values/Key	Char255Type	0..1
EFBPS/BankSvcRs/PmtInqRs/PmtAdviceRs/AdditionInfo/Values/Value	String	0..1

- **MsgCode**

Name	Value
MessageCode	PMTINQRS

- **Status Codes**

Code	Description
0	Payment Advice Accepted
9001	Invalid Sender Code
9002	Invalid Receiver Code
9003	Invalid Message Code
9004	Repeated RqUID
9005	Invalid Proxy Customer ID's Official ID Type
9006	Unable to Match Payment With Existing Bill
9007	Repeated PmtId
9008	Invalid Cust Id's Official Id Type
9009	PayAmt's Sequence Not Matched With An Existing Bill's Currency Amount
9010	Missing Exact Payment Currency Amount
9011	PayAmt Can't Be Zero
9012	For Range Payment Mode, Payment Amount Should Be Greater Than or Equal To The Minimum Amount
9013	For Range Payment Mode, and Maximum Amount Not Equal To Zero, Payment Amount Should Be Less Than or Equal Maximum Amount
9014	For Installment Payment Mode, Payment Amount Should Be Multiple of The Installment Amount or Equal To The Amount Due
9015	PayAmt Curcode Not Matched With Original Amount Due Master
9016	Not Valid Fees Amount
9017	Not Valid Fees Currency
9018	Invalid PrcDt, It Must be Grater Than or Equal The Current Business Date
9019	Invalid Bank Id
9020	Invalid Branch Code
9021	Invalid Access Channel Code
9022	Invalid PmtMethod Code
9023	Invalid Check Digit Value
9024	Invalid Service Type Code
9025	Invalid PmtRefInfo
9026	No Valid Bank Settlement Accounts
9027	Bill Status Does Not Allow Payment
9028	Bill In Hold State, Waiting For Biller Update
9029	Bill Had Expired, or Not Yet Valid For Payment
9030	The payments legnth is not valid
9031	Repeated Sequence
9032	Invalid Pmt Id Type Code
9033	Invalid Bill Number
9034	Invalid Billing Account
9035	Invalid Biller Id
9036	Invalid Message Code
9037	Bill already Paid
9038	Fractions is not allowed
9039	Invalid Payment Sequence
9040	The Customer Payment Name Is Required
9041	Payment In progress

5.1.6. Cancel Payment

Cancel Payment allows banks to Cancel successful payments.

5.1.6.1. Cancel Payment Request (CancelPmtRq message)

Element	Type	Cardinality
EFBPS	<EFBPS>	1
EFBPS/SignonRq	<SignonRq>	1
EFBPS/SignonRq/ClientDt	Date	1
EFBPS/SignonRq/LanguagePref	Open Enum	0..1
EFBPS/SignonRq/SignonProfile	<SignonProfile>	1
EFBPS/SignonRq/SignonProfile/Sender	Identifier	1
EFBPS/SignonRq/SignonProfile/Receiver	Identifier	0..1
EFBPS/SignonRq/SignonProfile/MsgCode	Identifier	1
EFBPS/BankSvcRq	<BankSvcRq>	1
EFBPS/BankSvcRq/RqUID	UUID	1
EFBPS/BankSvcRq/CancelPmtRq	< CancelPmtRq >	1
EFBPS/BankSvcRq/ CancelPmtRq /EPayBillRecID	UUID	1
EFBPS/BankSvcRq/ CancelPmtRq/PmtRec	<PmtRec>	1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtTransId	<PmtTransId>	1..3
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtTransId/PmtId	NC-32	1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtTransId/PmtIdType	Closed Enum	0..1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/CustId	<CustId>	0..1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/CustId/OfficialId	C-32	1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/CustId/OfficialIdType	Open Enum	0..1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo	<PmtInfo>	1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/PayAmt	<CurAmt>	1..*
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/PayAmt/Sequence	NC-2	1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/PayAmt/Amt	Currency	1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/PayAmt/CurCode	NC-3	1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/FeesAmt	<PayAmt>	0..*
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/FeesAmt/Amt	Currency	1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/FeesAmt/CurCode	NC-3	1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/PrcDt	Date	1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/BillCycle	NC-16	0..1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/BillNumber	C-32	0..1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/AccountId	<AccountId>	1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/AccountId/BillingAcct	C-32	1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/AccountId/BillerId	NC-4	0..1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/BankId	NC-8	0..1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/DistrictCode	NC-10	0..1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/BranchCode	NC-10	0..1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/AccessChannel	Open Enum	0..1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/PmtMethod	Open Enum	0..1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/ServiceType	Open Enum	0..1
EFBPS/BankSvcRq/ CancelPmtRq /PmtRec/PmtInfo/PmtRefInfo	NC-80	0..1
EFBPS/BillerSvcRq/ CancelPmtRq /PmtRec/PmtInfo/AdditionInfo	<AdditionInfoType>	0..1
EFBPS/BillerSvcRq/ CancelPmtRq /PmtRec/PmtInfo/AdditionInfo/Values	<ValuesType>	1
EFBPS/BillerSvcRq/ CancelPmtRq /PmtRec/PmtInfo/AdditionInfo/Values/ Key	Char255Type	1
EFBPS/BillerSvcRq/ CancelPmtRq /PmtRec/PmtInfo/AdditionInfo/Values/ Value	String	1
EFBPS/BillerSvcRq/ CancelPmtRq/CancelReason	Cancel Reason Enum	1

- **MsgCode**

Name	Value
MessageCode	CNLPMTRQ

- **Status Codes**

Code	Description
0	Cancel Request Accepted
9050	Transaction not found
8999	General Error
9001	Invalid Sender Code

9002	Invalid Receiver Code
9003	Invalid Message Code
9004	Repeated RqUID
9005	Invalid Proxy Customer ID's Official ID Type
9006	Unable to Match Payment With Existing Bill
9007	Repeated PmtId
9008	Invalid Cust Id's Official Id Type
9009	PayAmt's Sequence Not Matched With An Existing Bill's Currency Amount
9010	Missing Exact Payment Currency Amount
9011	PayAmt Can't Be Zero
9012	For Range Payment Mode, Payment Amount Should Be Greater Than or Equal To The Minimum Amount
9013	For Range Payment Mode, and Maximum Amount Not Equal To Zero, Payment Amount Should Be Less Than or Equal Maximum Amount
9014	For Installment Payment Mode, Payment Amount Should Be Multiple of The Installment Amount or Equal To The Amount Due
9015	PayAmt Curcode Not Matched With Original Amount Due Master
9016	Not Valid Fees Amount
9017	Not Valid Fees Currency
9018	Invalid PrcDt, It Must be Grater Than or Equal The Current Business Date
9019	Invalid Bank Id
9020	Invalid Branch Code
9021	Invalid Access Channel Code
9022	Invalid PmtMethod Code
9023	Invalid Check Digit Value
9024	Invalid Service Type Code
9025	Invalid PmtRefInfo
9026	No Valid Bank Settlement Accounts
9027	Bill Status Does Not Allow Payment
9028	Bill In Hold State, Waiting For Biller Update
9029	Bill Had Expired, or Not Yet Valid For Payment
9030	The payments legnth is not valid
9031	Repeated Sequence
9032	Invalid Pmt Id Type Code
9033	Invalid Bill Number
9034	Invalid Billing Account
9035	Invalid Biller Id
9036	Invalid Message Code
9037	Bill already Paid
9038	Fractions is not allowed
9039	Invalid Payment Sequence
9040	The Customer Payment Name Is Required

5.1.7. SOAP web service interface

The messages are exchanged based on a SOAP based web service interface that carry on the below exchanged XML based messages

5.1.7.1. Web service operations

- **enquireBills**

  The enquireBills operation will carry the BillInqRq and BillInqRs XML messages

- **confirmPayment**

  The confirmPayments operation will carry the PmtAdviceRq and PmtAdviceRs XML messages

- **calculateCommission**

  The calculateCommission operation will carry the FeeInqRq and FeeInqRs XML messages

- **getBillers**

  The getBillers operation will carry BillersInqRq and BillersInqRs XML messages

- **enquirePaymentStatus**

  The enquirePaymentStatus operation will carry PmtInqRq and PmtInqRs XML messages

- **CancelPayment**

  The CancelPayment operation will carry CancelPmtRq XML message.

5.1.7.2. Web service paramters

Every operation conatins two main paramters, Request and Response
Request:

senderID: The bank code provided by e-Pay system
signature: The hash value of the xml message(UTF-8) encrypted by the bank private key(PKI) using signature algorithm SHA1withRSA
message: The xml message

Response:

receiverID: The bank code provided by e-Pay system
signature: The hash value of the xml message(UTF-8) encrypted by the e-Pay private key(PKI) using signature algorithm SHA1withRSA
message: The xml message

5.1.7.3. Web service fault codes

If the e-Pay system not able to process the received message, a SOAP fault will be generated with the below fault codes.

Code	Description
0001	Invalid Sender ID
0002	Inavlid Message Signature
0003	Inavlid Message Format
0004	Invalid Message Against XML Schema

6. Common Aggregates

6.1. Enums

6.1.1. AccessChannel

The AccessChannel enum represents the application with which humans interface with EPAY. The value is an Open Enum as new channel types will continue to evolve and be introduced to the cutomers.

Code	Description
ATM	Bank ATM
IVR	Interactive Voice Recognition System
KIOSK	Bank Kiosk
INTERNET	Internet Browser
PORTAL	EPAY's Portal for Bank/Biller Interfacing
BTELLER	Bank Teller
POS	Point of Sale
DDS	Direct Debit Service

6.1.2. AccessChannel

This is an Open Enum data type to permit the addition of account types specific to a financial institution, country, etc. New enum values do not need to be registered with EPAY other than when defining the accounts to which EPAY must issue transfer instructions.

Code	Description
CUA	Current Account
CMA	Cash Management Account
DDA	Demand Deposit Account
LOC	Consumer Line of Credit
MMA	Money Market Account
SDA	Savings Account

6.1.3. BillCategory

A Bill Category assigns behaviour to a bill, thus permitting EPAY to apply rules to bills in varying manners. For example, a normal recurring bill requires different behaviour at time of presentment and payment than an endorsement bill.

Billers must assign category strings of their choosing to each of their bills, but must first register these strings with EPAY to avoid rejection.

Code	Description
BBP	Bill Based Payment

For the bill upload request, the bill category will be validated against the biller configuration profile.

6.1.4. BillStatusCode

The BillStatusCode identifies a bill's status within EPAY or Biller systems and is intended to be used when communicating bill data between disparate systems.

6.1.4.1. BillStatusCode – Bill Presentment

Code	Description
BillPaid	Fully Paid
BIllPartialPd	Partially Paid
BillOverPd	Over Paid
BillUnpaid	No payments have been applied to this bill
BillHolded	Pending payments has not been confirmed by biller
BillClosed	Biller not allowing payment for this bill
PmtNotAllowed	Bill payment not allowed by service provider

6.1.4.2. BillStatusCode – Bill Upload

Code	Description
BillNew	Bill record is new, no corresponding bill exists in the system.
BillUpdated	Bill is an updated version of a bill currently in the EPAY system.
BillAny	EPAY makes no restrictions regarding the existence of a bill in the system, EPAY will perform either BillNew or BillUpdate
BillExpired	The specified bill should be set to "expired".

6.1.5. BillSummAmtCode

The Bill Summary Amount Code enum adds a descriptive element to a currency amount found in a BillSummAmt aggregate

Code	Description
Incentive	The CurAmt in the corresponding BillSummAmt record contains an incentive amount contained in the bill record's amount due. The corresponding ShortDesc aggregate may contain a localized string describing the reason for the incentive.
Penalty	The CurAmt in the corresponding BillSummAmt record contains a penalty amount contained in the bill record's amount due. The corresponding ShortDesc aggregate may contain a localized string describing the reason for the penalty.
CreditLimit	Assigned to SummSummAmt records with a type of "InfoOnly" to convey an upper limit value which a bill may accumulate before services are deactivated.

6.1.6. BillSummAmtType

Code	Description
Payable	The CurAmt in the BillSummAmt aggregate represents a payable amount. Many payable amounts may exist in the same bill record. The CurAmt in the BillSummAmt aggregate represents a currency amount
Supplemental	which is included in the bill record's payable amount, but cannot be separately paid.
InfoOnly	The corresponding BillSummAmt represents optional information pertaining to the bill.

6.1.7. LanguagePref

Code	Description
ar-eg	Egypt dialect for Arabic
en-gb	Great Britain dialect for English

6.1.8. OfficialIdType

An OfficialId is a unique identifier for a bill-paying entity, including people and companies. There is a number of possible identification schemes used to identify bill payers in EPAY.

Code	Description
NID	National ID
SID	Social ID
PSN	Passport Number
BTL	Bank Teller Number
BEN	Bank Employee Number
BIE	Biller Employee Number
EEN	EPAY Employee Number

6.1.9. PmtIdType

PmtId represents a Payment Transaction Number. PmtIdType determines whether this transaction number has been provided by the Bank, Biller, or EPAY.

Code	Description
EPTN	EPAY Payment Transaction Number
BLRPTN	Biller Payment Transaction Number
BNKPTN	Bank Payment Transaction Number

6.1.10. PmtMethod

Code	Description
CASH	Cash
CCARD	Credit Card
EFT	Electronic Funds Transfer
ACTDEB	Debit to Account

6.1.11. ServiceType

A ServiceType represents the solitary service category associated with a bill. The intent is to permit Customers to query on bills according to the type of service for which they are being charged. It also permits EPAY to associate messages with bills according to the associated service type.

Code	Description
ELCT	Electricity
PHON	Phone
INSR	Insurance
BKSV	Bank Service
GOVT	Government Service
MED	Medical
CCRD	Credit Card
EUP	Egyptian Universities Payments
CAU	Cairo University Payment

6.1.12. Cancel Reason

Cancel reason represents the reason of taking cancel decision and send the cancel request

Code	Description
1	Payment Advice not completed
2	Payment Advice not updated
3	Payment Advice timeout

6.2. Common Aggregates

6.2.1. AccountId Aggregate

This aggregate represents a bill presentment account. Each account is identified by a unique account number which is provided by the Biller, Billing Account must be unique per Biller Id

Element	Type	Cardinality
BillingAcct	C-32	1
BillerId	NC-4	0..1

The BillingAcct value represents the actual account number in a Biller's system and is the primary value within the AccountId aggregate.

A Biller is always identified in EPAY messages using the unique identifier assigned to it by EPAY; The BillerId element represents this value.

For the upload request, if the sender is the biller, the BillerId is not required to be repeated for each account.

6.2.2. BillInfo Aggregate

The BillInfo aggregate is used to define a collection of values assigned to a specific bill. It is difficult to define a single data structure, however, to meet the needs of all Billers.

Element	Type	Cardinality
BillCategory	Open Enum	1
ServiceType	Open Enum	1
BillCycle	NC-16	0..1
BillNumber	C-32	0..1
AccountId	<AccountId>	1
AmountDue	<CurrAmt>	1..*
DueDt	Date	1
OpenDt	Date	0..1
CloseDt	Date	0..1
ExpDt	Date	0..1
ChkDigit	NC-2	0..1
BillRefInfo	NC-80	0..1
BillSummAmtGroup	<BillSummAmtGroup>	0..*

6.2.3. BillCategory Element

The BillCategory enum is used to help uniquely identify a bill, Bills are typically identified by Account/BillNumber or Account/BillCycle. The identification mechanism depends on the business rule defined for the bill's category.

6.2.3.1. ServiceType

The ServiceType enum defines the type of service provided by the Biller and associated with a given bill.

6.2.3.2. BillNumber Element

BillNumber is used by several bill categories as a unique bill identifier within the account, but may actually be NULL for many bills. The value is required when several bills of the same category and bill cycle can be associated with the same account. In such a scenario, this unique identifier serves as the only tool for identifying the intended bill.

6.2.3.3. BillCycle Element

The BillCycle element identifies a Biller's billing cycle and is often used in conjunction with an account identifier for uniquely identifying a bill as most bill categories permit only one bill of each category with a specific account during each billing cycle. The format of the BillCycle element is specified by the Biller, but cannot exceed 16 characters.

6.2.3.4. BillingAcct Element

The bill's associated account number is contained in the BillingAcct element.

6.2.3.5. BillerId Element

The bill identifier code is conveyed in the BillerId element. The value is optional to accommodate the bill upload scenario where it is possible an entire file of several hundred thousand records all share the same Biller. In this scenario, the BillerId is conveyed in the file's signon aggregate.

6.2.3.6. Various Date Elements

The Biller is permitted to assign several date values to each bill:

- The CloseDt value indicates when the bill was frozen and prepared for presentation.
- OpenDate marks the start date for assigning charges to a bill.
- The SettlDt element defines the date on which a bill is considered past due, Payments are typically considered late if submitted to EPAY beyond this date.
- An expiration date is optional and indicates when the bill is no longer considered valid, this value is ideal for scenarios where a bill is considered an offer to purchase goods (for example, an insurance endorsement) if the purchase is made by a specified date.

6.2.3.7. ChkDigit

The Biller may choose to upload a check digit string with each bill. This string is identical to the check digit string provided on the printed bill delivered to the Customer and is used to validate the exact payment amount. Business rules may be implemented to validate against this value.

6.2.3.8. BillRefInfo Element

Some Billers prefer payments be submitted with an additional reference string associated with the paid bill, The BillRefInfo element represents this optional value and can be used to reference a Biller specific value which is subsequently attached with associated payments.

6.2.3.9. BillSummAmtGroup

Billers may use the nested BillSummAmtGroup aggregate to show an informational breakdown of the bill. A credit limit value, incentive, or penalty amounts are useful implementations of this aggregate.

It is important to note these key / value pairs are for information only and in no way are used in calculations to derive a bill's amount due. Billers may choose to ignore this aggregate without any ill effects within the Bill.

6.2.4. *BillInfoError Aggregate*

This aggregate is simply a wrapper around a BillInfo aggregate's primary values and is used to communicate errors associated with bill data. This aggregate is intended almost exclusively to report error conditions back to the Biller during the Bill Upload process.

Element	Type	Cardinality
Error	<Error>	1
AccountId	<AccountId>	1
BillCategory	Open Enum	0..1
ServiceType	Open Enum	0..1
BillCycle	NC-16	0..1
BillNumber	C-32	0..1

6.2.4.1. Error Aggregate

The Error aggregate informs the Biller of the actual error which occurred during bill upload. The ErrorMsg element contains an English text string description of the problem. Billers should rely primarily upon the error code to identify and troubleshoot problems.

6.2.4.2. Bill Details

The remainder of the aggregate contains sufficient details for the Biller to uniquely identify the rejected bill record. These data points include account number, bill cycle, bill number, bill category, and service type.

6.2.4.3. BillPmtStatus Aggregate

This aggregate is almost identical to the PmtStatus aggregate, but adds additional context of an associated payment record. As such, the aggregate does not need to be embedded within a PmtRec aggregate.

Element	Type	Cardinality
PmtTransId	<PmtTransId>	1..3
PmtStatusCode	Open Enum	1
CurAmt	<CurAmt>	1
EffDt	Date	0..1

6.2.4.4. PmtTransId

The PmtTransId element represents a unique Payment Transaction Number. This number can be either the one assigned by Bank, EPAY, or the Biller, depending on the value of the PmtIdType element.

6.2.4.5. PmtStatusCode

The PmtStatusCode conveys context sensitive information which is relevant to the related process.

6.2.4.6. EffDt and CurAmt

EffDt and CurAmt represent the payment date and amount. These values are also required if this aggregate is used to load payment data.

6.2.5. BillPmtStatusError

The BillPmtStatusError aggregate is a wrapper which associates an Error aggregate with a single instance of a BillPmtStatus aggregate. The intent is to communicate errors encountered while processing individual BillPmtStatus instances.

Element	Type	Cardinality
Error	<Error>	1
BillPmtStatus	<BillPmtStatus>	1

6.2.5.1. Error

The Error aggregate conveys the cause of a processing error with a BillPmtStatus aggregate. The proper error codes are provided with message definitions utilising this aggregate.

6.2.5.2. BillPmtStatus

This BillPmtStatus aggregate is an echo of the corresponding BillPmtStatus aggregate in the message communicating the payment information.

6.2.6. BillRec Aggregate

The BillRec aggregate wraps the BillInfo data to add such context as the bill's owner(s), Biller associated payment status', and duplicate bill data. In other words, the aggregate provides additional bill-specific data which may be associated with a given bill. The aggregate is most commonly used during the Bill Upload and Inquiry processes.

Element	Type	Cardinality
SupercedeBillNumber	C-32	0..1
BillStatusCode	Closed Enum	0..1
BillPmtStatus	<BillPmtStatus>	0..*
CustId	<CustId>	0..*
DisplayLabel	<DisplayLabel>	0..*
Msg	<Msg>	0..1
BillInfo	<BillInfo>	1

6.2.6.1. SupercedeBillNumber

The SupercedeBillNumber element is only used in scenarios where the BillInfo's BillNumber value is required to be non-NULL and is used to identify duplicate or outdated bills. For example, some Billers permit multiple bills of the same category to be associated with an account during any given billing cycle. An attempt to update a previously loaded bill with new data will fail unless the upload file specifies which of these bills is be modified.

6.2.6.2. BillStatusCode

BillStatusCode is an enum intended for use during the Bill / Payment Presentment process, and informs the Bank application whether the bill is unpaid, partially paid, fully paid or overpaid. This enum is also used during the Bill Upload process to expire / cancel bills, deactivate accounts, and other similar tasks.

6.2.6.3. BillPmtStatus

BillPmtStatus, however, identifies which payments are associated with the referenced bill and is used during the Bill Upload process.

6.2.6.4. CustId

Billers may choose to have construct Customer Profiles by associating Customers with bills during the Bill Upload process. This association is accomplished using the CustId repeating aggregate.

6.2.7. CurrAmt Aggregate

The CurrAmt aggregate will be used in both the payment details, and the bill info details, in case of bill info details, it will represents the bill's due currency amounts, while in case of payment details, it will represent the amounts which have been paid for a specific bill info amount due.

The aggregate is mandatory, and can be repeated for different payment modes and is being used by the EPAY for validation during the payment process.

Element	Type	Cardinality
Sequence	NC-2	1
OriginalAmt	Currency	1
AmtDue	Currency	1
CurCode	NC-3	1
ExactPmt	Boolean	1
PymtMode	C-32	0..1
RangeInfo	< RangeInfo>	0..1
RangeInfo/MinAmt	Currency	0..1
RangeInfo/MaxAmt	Currency	0..1
InstInfo	< InstInfo>	0..1
InstInfo/NumInst	NC-2	1
InstInfo/InstAmt	Currency	1
InstInfo/MinInst	NC-2	1
ShortDesc	<ShortDesc>	0..*
SettleAccCode	C-32	0..1

6.2.7.1. Sequence

The Sequence element represents the unique identifier of the CurrAmt aggregate, and will be used during the payment upload to match the bill amount due with the current bill info amount due stored in the bill store, and also will be used in both payment advice and payment notification, for matching the customer paid amount to the bill info amount due, to validate that it follows the limitations defined by the payment mode element.

6.2.7.2. AmtDue

The AmtDue element represents a currency value which is part of the bill's remaining due amount; the customer didn't pay until now, and is asked to pay, for the bill infor details the

amount due represents the amount the customer is asked to pay, while in the payment details, the amount due respresents the payment paid amount.

6.2.7.3. OriginalAmt

The OriginalAmt element represents a currency value which is part of the bill's original due amount at the biull issuance time, and the customer is asked to pay now, this elements is mandatory for the bill infor details, while for the payment details it is not required.

6.2.7.4. CurCode

The CurCode element represents the currency in which the customer is asked to pay the remaining amount due.

6.2.7.5. ExactPmt

The ExactPmt element indicate by the biller however this amount due must be paid or an optional element.

6.2.7.6. PymtMode

The PymtMode element is used by the biller to define the value range, in which the customer is requested to pay through the bank interface, and system will reject any payment advice sent from the bank side not in the allowed range, the element is mandatory in case of bill info details, the following is payment mode allowed values.

Code	Description
Installment	The requested amount due can be paied as installments, and the customer can pay only a fixed number of installments, not amount range allowed.
Range	The requested amount due can be paied with a min and max amount.

6.2.7.7. RangeInfo element

The RangeInfo element is mandatory in case if the selected payment mode is "Range", and defines the maximum and minimum amount which can be paid by the customer.

In case the MinAmt and MaxAmt elements are both zeros, the customer is allowed to pay any amount without any limitation even if it exceeds the amount due.

6.2.7.8. InstInfo element

The InstInfo element is mandatory in case if the selected payment mode is "Installment", and define the number of installments in which this amount due can be paid.

The InstAmt carry the installment amount, and the customer can only pay a multiple of the mentioned installment amount, and in case the customer would like to pay all the installments, he will be asked to pay the Amt element value not the multiples of the MinAmt element.

The MinInst represents the minimum number of installments in which the customer can pay.

The bank must round the paid amount to the min units of the paid currency, i.e for EGP the fraction will be only 2 digits.

6.2.7.9. ShortDesc

ShortDesc is a localised string describing the value and is intended only for human consumption. The Biller is encouraged to upload both Arabic and English versions of these descriptive strings to ensure satisfactory display of these optional data points, the element is mandatory in case of bill info details.

6.2.7.10. SettleAccCode

The SettleAccCode element is being used by the Biller during the bill info upload to define the account which will be used by the EPAY system to settle the amount which will be paid by the customer for this specific CurrAmt Aggregate, the account info which will be used during the settlement process must be preconfigured in the EPAY system, the EPAY operation team will be responsible to acknowledge the biller with the configured account code, the element is mandatory in case of bill info details biller upload only.

6.2.8. BillSummAmtGroup Aggregate

The BillSummAmtGroup Aggregate represents a group of BillSummAmt elements which are related to a specific description the biller would like to represent ot the customer.

Element	Type	Cardinality
GroupSeq	Currency	1
ShortDesc	<ShortDesc>	1..*
BillSummAmt	< BillSummAmt>	1..*

6.2.8.1. GroupSeq

The GroupSeq element define the order in which group can be presented to the customer.

6.2.8.2. ShortDesc

ShortDesc is a localised string describing the group and is intended only for human consumption. The Biller is encouraged to upload both Arabic and English versions of these descriptive strings to ensure satisfactory display of these optional data points.

6.2.8.3. BillSummAmt

The BillSummAmt Aggregate the list of currency amounts that are distinctly represented in the bill's due amount.

6.2.9. BillSummAmt Aggregate

The BillSummAmt aggregate represents any currency amounts that are distinctly represented in the bill's due amount. The aggregate is purely optional and is not used by EPAY for any other purpose than display. The values in this aggregate do not need to add up to the associated bill's due amount.

Element	Type	Cardinality

Amt	Currency	1
CurCode	NC-3	1
CurRate	Currency	0..1
BillSummAmtCode	Open Enum	0..1
BillSummAmtType	Open Enum	1
ShortDesc	<ShortDesc>	0..*

6.2.9.1. Amt & CurCode

The Amt & CurCode elements represents a currency value the Biller claims is already incorporated into a bill's due amount.

6.2.9.2. CurRate

The CurRate element will be used by the Biller to define to the customer the rate which has been used to convert this currency amount of this BillSummAmt to the amount due currency amount, this element is optional in case of same currency collection.

6.2.9.3. BillSummAmtCode and BillSummAmtType

The BillSummAmtCode and BillSummAmtType describe the currency value on a functional level.

6.2.9.4. ShortDesc

ShortDesc is a localised string describing the value and is intended only for human consumption. The Biller is encouraged to upload both Arabic and English versions of these descriptive strings to ensure satisfactory display of these optional data points.

6.2.10. CustId, ProxyCustId Aggregate

The CustId aggregate identifies a specific bill paying Customer or institution. Such identification is accomplished using a National ID, Social ID, Business ID, EPAY ID, or other value.

Element	Type	Cardinality
OfficialId	C-32	1
OfficialIdType	Open Enum	0..1

6.2.10.1. OfficialId

The OfficialId element contains the actual identifier value which uniquely identifies a person or business via some specific identification system. This value is typically a person's National ID, but could be some other value.

6.2.10.2. OfficialIdType

The OfficialIdType element declares the type of value that is contained in the OfficalId element. By default, EPAY assumes the use of NationalId.

6.2.11. DateRange Aggregate

The DateRange aggregate defines a contiguous range of days by specifying both a start and end date value.

Element	Type	Cardinality
StartDt	Date	0..1
EndDt	Date	0..1

StartDt represents the older date, whereas EndDt represents the newer time. It is possible to leave the date range open-ended by leaving one of the two date values as NULL. It is invalid, however, to leave both the Start and End date elements NULL.

6.2.12. Error Aggregate

This aggregate is simply a wrapper around a BillInfo aggregate's primary values and is used to communicate errors associated with bill data. This aggregate is intended almost exclusively to report error conditions back to the Biller during the Bill Upload process.

Element	Type	Cardinality
ErrorCode	Long	1
ErrorMsg	C-255	1

6.2.12.1. ErrorCode Element

The ErrorCode value informs the Biller of the actual error which occurred during bill upload.

6.2.12.2. ErrorMessage

The ErrorMsg element contains an English text string description of the problem. Billers should rely primarily upon the error code to identify and troubleshoot problems.

6.2.13. DisplayLabel

The DisplayLabel aggregate permits the Biller to provide a short description of each bill during the Bill Upload process. The Bill Query process either derives the display label value from business rules or extracts the value directly from uploaded data.

Element	Type	Cardinality
LanguagePref	Open Enum	0..1
Text	C-32	1

6.2.13.1. LanguagePref

LanguagePref specifies the localised language of the message. Only Arabic and English are supported.

6.2.13.2. Text

This localised string provides a short description of a bill. The value is always derived from Biller data and should not be confused with Bank aliases.

6.2.14. Msg Aggregate

The Msg aggregate communicates a localised text string the message recipient, and is primarily intended to associate Biller- or EPAY-based messages with bill inquiry results. The aggregate is sufficiently abstract, however, to permit other arbitrary usages.

Element	Type	Cardinality
LanguagePref	Language	1
Text	C-4096	1

6.2.14.1. LanguagePref

LanguagePref specifies the localised language of the message. Only Arabic and English are supported.

6.2.14.2. Text

Text is a potentially long text string containing the localised message.

6.2.15. PmtInfo Aggregate

The PmtInfo aggregate represents a payment which has taken place.

Element	Type	Cardinality
CurAmt	<CurAmt>	1
PrcDt	Date	1
SettlDt	Date	0..1
BillCycle	NC-16	0..1
BillNumber	C-32	0..1
AccountId	<AccountId>	1
BankId	NC-8	0..1
DistrictCode	NC-10	0..1
BranchCode	NC-10	0..1
AccessChannel	Open Enum	0..1
PmtMethod	Open Enum	0..1
ChkDigit	NC-2	0..1
ServiceType	Open Enum	0..1
PmtRefInfo	NC-80	0..1

6.2.16. PmtInfoError Aggregate

This aggregate serves as a logical wrapper for the PmtRec aggregate, but includes an error code and message.

Element	Type	Cardinality
Error	<Error>	1
PmtTransId	<PmtTransId>	0..*

The PmtInfo aggregate is similar to the PmtRec aggregate. In fact, the PmtRec aggregate actually wraps a single instance of PmtInfo. The PmtInfo aggregate is used only to describe a payment which may exist, whereas the PmtRec aggregate supplements the PmtInfo aggregate with additional data points for payments which have actually taken place.

6.2.16.1. CurAmt Element

The CurAmt element contains the payment amount.

6.2.16.2. PrcDt and SettlDt Elements

PrcDt is equivalent to the transaction date, and therefore is used to represent the date and time on which the Customer initiated the payment transaction. SettlDt represents the date where the payment will be settled.

6.2.16.3. Bill Identification Elements

BillerId, AccountId, BillNumber, and BillCycle are used to collectively define the bill against which a payment is to be made. The manner in which these elements are used is identical to the rules in which the BillInfo aggregate uses these elements for unique bill identification.

6.2.16.4. BankId

This value is the fullfillment entity code assigned to each EPAY-participating Bank.

6.2.16.5. PmtMethod

Each payment has an associated payment amount and method. The PmtMethod enum determines the method of payment (for example, cash, credit card, etc.).

6.2.16.6. ChkDigit

The ChkDigit value is that which was entered by either the Customer or ProxyCustomer when submitting payment. EPAY may validate this string against payment value if Biller-profile configured for that, otherwise the value is passed through to the Biller for validation.

6.2.16.7. ServiceType

The ServiceType value represents the type of service against which the payment has been submitted.

6.2.16.8. PmtRefInfo Element

The PmtRefInfo element represents a Biller-defined text which must be included with the payment. The content of this field is copied from the BillRefInfo element of the bill against which this payment is to be made. It is possible for this value to be used when matching payments to bills.

6.2.16.9. Branch and District Codes

EPAY will maintain the Bank and branch codes provided by the Bank at the time of payment. The access channel used by the Customer for payment initialisation is included in this message, as is the method of payment.

6.2.16.10. Error Aggregate

The Error aggregate indicates the cause of payment rejection. The ErrorMsg element is a brief English-only description of the error and is not intended for Customer display. The ErrorMsg string is typically a technically-oriented string intended for troubleshooting. Bank and Biller

applications should rely upon the ErrorCode to determine what message to convey to Customers as the ErrorMsg element is not appropriate for human display.

6.2.16.11. PmtTransId

The PmtInfoError aggregate only wraps the PmtTransId fragment of the PmtInfo aggregate as this should be sufficient for the receiving application to uniquely identify the rejected payment record. The PmtTransId is a repeating aggregate, where each instance of the aggregate represents the same payment, but with a different transaction number. For example, it is possible to embed both EPAY and Biller Payment Transaction Number for the same payment.

6.2.17. PmtRec Aggregate

The PmtRec aggregate differs from PmtInfo in that the former represents an actual payment whereas the latter represents a desire to make a payment. In fact, the PmtRec aggregate is actually a wrapper for the PmtInfo aggregate which adds Customer, Payment Status and Transaction Numbers to the PmtInfo aggregate.

Element	Type	Cardinality
PmtTransId	<PmtTransId>	1..*
CustId	<CustId>	0..1
PmtStatus	<PmtStatus>	1
PmtInfo	<PmtInfo>	1

6.2.17.1. PmtTransId

Note that repeating the PmtTransId aggregate permits multiple Payment Transaction Numbers for a single payment. This is possible as each EPAY-participating organisation may wish to assign their own transaction number to an individual payment record.

It is an error to have two or more PmtTransId aggregates with the same PmtIdType enum value as there can only be one transaction number of each type for each payment. It is rare that only one Payment Transaction Numbers are assigned to a PmtRec, but may occur during payment upload as the Biller may choose not to provide their transaction numbers. EPAY assigns a Payment Transaction Number to each payment during the upload process, as do Banks during the advice message. So it is assured that all other contexts in which the PmtRec aggregate is used will have at lease one assigned Payment Transaction Number.

6.2.17.2. CustId

CustId represents the individual who submitted the payment.

6.2.17.3. PmtStatus

The PmtStatus aggregate identifies the payment’s lifecycle state.

6.2.17.4. PmtInfo

The solitary PmtInfo aggregate wrapped by PmtRec identifies required payment attributes in a context insensitive manner.

6.2.18. PmtRecAdviceStatus

This aggregate is used to provide status for individual payment payments during the Payment Advice process. During this process, the Bank sends a Payment Advice message with one or more PmtRec aggregates. Each such aggregate must contain a Bank Payment Transaction Number.

Element	Type	Cardinality
PmtTransId	<PmtTransId>	1..2
Status	<Status>	0..1

6.2.18.1. PmtTransId

This aggregate uses PmtTransId to uniquely identify a specific payment. Either one or two instances of this aggregate will exist. The required instance represents the EPAY Payment Transaction Number. Advice messages representing successfully transferred Customer funds must also include the Bank's Payment Transaction Number as the second PmtTransId value. Failed payments will not include this value.

6.2.18.2. Status

This aggregate is only required for unsuccessful funds transfers and indicates the cause of failure. A missing Status aggregate implies successful funds transfer for the associated payment.

6.2.19. PmtStatus Aggregate

The PmtStatus aggregate communicates a payment's lifecycle status. The aggregate is always used in the context of a PmtRec aggregate.

Element	Type	Cardinality
PmtStatusCode	Closed Enum	1
EffDt	Date	0..1

6.2.19.1. PmtStatusCode

This value represents the payment's current lifecycle state.

6.2.19.2. EffDt

The EffDt value is optional and specifies the date in which the payment's current lifecycle state was reached. This value contains both date and time values.

6.2.20. PmtTransId Aggregate

This aggregate contains both a Payment Transaction Number and an identifier defining the organization which has assigned the number.

Element	Type	Cardinality
PmtId	NC-32	1
PmtIdType	Closed Enum	0..1

6.2.20.1. PmtId

PmtId represents an organisation transaction number.

6.2.20.2. PmtIdType

This value defines the owner of the Payment Transaction Number.

6.2.21. ShortDesc Aggregate

The ShortDesc aggregate provides a generic mechanism for various messages and aggregates to contain localised text strings.

Element	Type	Cardinality
LanguagePref	Language	1
Text	C-255	1

6.2.22. Signature Aggregate

The Signature aggregate carries a digital signature for integrity, authenticity and non-repudiation of the message. The aggregate contains the X-Path of the signed data and base-64 encoded signature value.

Element	Type	Cardinality
XPath	C-256	1
SignatureValue	C-∞	1

6.2.23. SignonProfile Aggregate

The SignonProfile is used in both the SignonRq and SignonRs aggregates to identify the message sender and the intended recipient. Only an Identifier type is required to identify either the sender or receiver as the organisation type of each entity is implied by the actual message.

Element	Type	Cardinality
Sender	Identifier	1
Receiver	Identifier	0..1
MsgCode	Identifier	1

6.2.23.1. Sender / Receiver Element

The Sender element represents the organisation sending the associated XML document, while The Receiver element represents the organisation receiving the associated XML document, and they are defined as follows:

- Bank: FE Code
- Biller: SP Code
- EPAY: Identified by the token “EPAY”

6.2.23.2. MsgCode Element

The MsgCode is a unique code representing the intent of the XML document.

6.2.24. SignonRq Aggregate

The SignonRq message is used to authenticate a client. The message is always located at the top of the document and never within a service wrapper. Improper authentication causes the associated messages to be rejected, and anonymous messages are not permitted.

Element	Type	Cardinality
ClientDt	Date	1
LanguagePref	Open Enum	0..1
SignonProfile	<SignonProfile>	1

6.2.24.1. ClientDt

The ClientDt represents the date and time the client application originally sent the associated document.

6.2.24.2. LanguagePref

The LanguagePref enum identifies the preferred human language for the current transaction, in case the customer didn't select his prefared language. The value is optional, however, as Arabic is always assumed unless otherwise specified.

6.2.24.3. SignonProfile

The SignonProfile aggregate identifies the sender of the document, the intended recipient, and other pertinent data required for authentication. This profile aggregate is required as anonymous signons are not supported.

6.2.25. SignonRs Aggregate

This aggregate is located at the top-level of all response messages and serves a dual role of both authentication and status report of the associated request. For security reasons, the <SignonRs> does not echo all elements of the request.

Element	Type	Cardinality
ClientDt	Date	0..1
ServerDt	Date	1
LanguagePref	Open Enum	0..1
SignonProfile	<SignonProfile>	1

6.2.25.1. ClientDt

The ClientDt represents the date and time the client application originally sent the associated document.

6.2.25.2. ServerDt

This element contains the date/time the response message was generated. This time is relative to the machine generating the response.

6.2.25.3. LanguagePref

The LanguagePref enum identifies the preferred human language for the current transaction. The value is optional, however, as Arabic is always assumed unless otherwise specified.

6.2.25.4. SignonProfile

The SignonProfile aggregate identifies the sender of the document, the intended recipient, and other pertinent data required for authentication. This profile aggregate is required as anonymous signons are not supported.

6.2.26. Status Aggregate

The Status aggregate is used in each response to indicate the status of the message. The actual status values depend on the response message. Valid response values are provided throughout this document.

Element	Type	Cardinality
StatusCode	Long	1
ShortDesc	C-255	0..1