



Owned by Oman national Engineering and Investment Company "ONEIC"

Con2Us

Integration Interface Technical
Details
ONEIC-TASDEED Project

Document Version: V.1.4.6
Date: May 1, 2017





ONEIC-TASDEED Switch

Integration Interface Technical Details

Version 1.4.6

COPYRIGHTS

Copyright© **ONEIC-TASDEED** 2017. All rights reserved.

No parts of this document may be copied or reproduced, even temporarily, in whole or in part, in any manner or form, or on any media, nor sold, licensed, distributed or transferred to any person, or used, adapted, translated, without the prior written permission of **ONEIC-TASDEED**.

The recipient is, however, authorized to copy or reproduce this publication within its own organization if necessary for the purpose for which it is supplied. Any such copy or reproduction will include the following: acknowledgement of the source, reference and date of publication, and all notices set out on this page.

DISCLAIMER

Although **ONEIC-TASDEED** has made every effort to make this document accurate, up-to date, and complete, **ONEIC-TASDEED** offers no warrants, express or implied, related to this document. In no event shall **ONEIC-TASDEED** be liable for any loss of profits, loss of business, loss of use or data, interruption of business, or for indirect, special, incidental, or consequential damages of any kind arising from any error in this document.



SEND US COMMENTS

ONEIC-TASDEED welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the revision process.

If you find any errors or have any other suggestions to improve the document quality and clarity, please indicate the chapter and page number (if available).

Please send comments to: support@madfooat.com

Document Control

Document Information

Item	Information
File Name	ONEIC-TASDEED_Integration_Interface_Con2Us_V.1.4.6
Document Version	V.1.4.6
Document Owner	ONEIC-TASDEED: Owned by Oman National Engineering and Investment Company, operated by ONEIC-TASDEED for Electronic Payments 'TFEP'
Issue Date	Jan 31, 2016
Last Saved Date	May 1, 2017



Con 2 Us

Integration Specs

		<p>Added Payer Info container in Biller Prepaid Validation Request (5.11), Added Payer Info container in Payment Notification Request(5.13), Added Payer Info container in Payment Acknowledgment Request(5.14), Added new element <NickName> into Account Inquiry Service (5.3) and Add customer billing service (5.4), Added the specification of <NickName> element into Services Elements Specs. (4.11), Added a new Operations Types (4.22.1), Added new Enums (4.22.21, 4.22.22), Added a new Service Error Code (4.22.19), Added a new business rules section related to Biller Service Availability (6.21), Added a new business rules section related to System Service Availability (6.22), minor amendments on business rules.</p> <p>Update the business rules in the following sections:</p> <p>Section 6.4 => Rule No. 4</p> <p>Section 6.5 => Rule No. 10</p> <p>Section 6.10 => Rule No. 15</p> <p>Section 6.10 => Rule No. 16</p> <p>Section 6.10 => Rule No. 20</p> <p>Section 6.10 => Rule No. 25</p> <p>Section 6.11 => Rule No. 2</p> <p>Section 6.11 => Rule No. 21</p> <p>Section 6.12 => Rule No. 5</p> <p>Section 6.12 => Rule No. 27</p> <p>Section 6.13 => Rule No. 10</p> <p>Section 6.13 => Rule No. 31</p>
--	--	--



Con 2 Us
 Integration Specs

			<p>Section 6.13 => Rule No. 32</p> <p>Section 6.14 => Rule No. 12</p> <p>Section 6.14 => Rule No. 14</p> <p>Section 6.14 => Rule No. 13</p> <p>Section 6.16 => Rule No. 19</p> <p>Section 6.16 => Rule No. 20</p> <p>Section 6.18 => Rule No. 14</p>
1.4.6	May 1, 2017	TFEP	<p>Modifications on the Enums sections:</p> <p>4.22.2, 4.22.12, 4.22.19</p> <p>Modifications on the business rules sections:</p> <p>6.10, 6.11, 6.13, 6.14, 6.16</p> <p>Modifications on the Appendix section:</p> <p>7.1</p>



C o n 2 U s
Integration Specs

Confidentiality

This document contains information confidential and proprietary to ONEIC-TASDEED. The information may not be used, disclosed or reproduced without the prior written authorization of ONEIC-TASDEED and those so authorized may only use the information for the purpose of evaluation consistent with authorization.

Reproduction of any section of this document must include this legend © ONEIC-TASDEED 2017.



Table of Contents

ONEIC-TASDEED Switch.....	1
Copyrights.....	1
Disclaimer	1
Send us comments.....	2
Document Control.....	2
Document Information.....	2
Document History	Error! Bookmark not defined.
Version Approvals	Error! Bookmark not defined.
Confidentiality	5
Table of Contents.....	6
Table of Figures	Error! Bookmark not defined.
Glossary.....	7
1. Introduction.....	17
1.1 Introduction.....	17
1.2 Document Purpose	17
1.3 Document Scope	17
1.4 Audience.....	17
1.5 Reference Documents	18
2. ONEIC-TASDEED Switch.....	19
2.1 Business Process Operations (BPOs) of ONEIC-TASDEED.....	20
2.2 Bill Upload Process.....	22
2.3 Bill Presentment (Bill Inquiry) Process	23
2.5 Payment Process (Postpaid & Prepaid Payments)	24
2.6 Settlement & Reconciliation Processes.....	26
3. Participants Preparations for ONEIC-TASDEED	26
3.1 Billers Preparations.....	27

Glossary

Term	Description
Access Channel	A channel where customers who owe bills used to pay through, most of these channels are Bank/PSP provided
Account Upload	A process that permits the Bank/PSP to manage (Add and Update) all accounts (customer profiles) in ONEIC-TASDEED system
Active Participant	A Participant that is allowed to communicate with ONEIC-TASDEED successfully
API	Application Programming Interface
Authentication	The process in which the participant should request a Token to be authorized a participant to be integrated with ONEIC-TASDEED switch and be able to perform any operation
Bank	Licensed banks operating in Jordan
Batch	A transaction that contains a bulk of records, such as: Bills
Bill Details	The information related to a bill, such as: due amount, due date, etc...
Bill Expiry Date	Represents the date of the bill when it is no longer available for payment over ONEIC-TASDEED network but still inquirable
Bill Inquiry	The process in which the Bank/PSP inquires ONEIC-TASDEED for a bill and associated payments data using a Bill Inquiry message
Bill Number	The biller's unique reference number for a bill



Con 2 Us
Integration Specs

Bill Payment	The process in which the Bank/PSP creates new payment record(s) in ONEIC-TASDEED
Bill Presentment	The same as Bill Inquiry; it is the process in which the Bank/PSP inquires ONEIC-TASDEED for a bill and associated payments data using a Bill Inquiry message, and
	ONEIC-TASDEED responds to that request by presenting the data to the Bank/PSP
Bill Status	Defines possible statuses of a bill within ONEIC-TASDEED and is intended to be used when communicating bill data between separated systems, such as: New, BillPartiallyPaid, Paid, etc...
Bill Type	Defines the possible types of a bill within ONEIC-TASDEED in terms of recurrence, such as: Recurring, OneOff
Bill Upload	The Bill Upload process permits the efficient transfer of bill data from Biller billing systems to ONEIC-TASDEED
Biller	A company or organization that generates bills legally that should be paid for its customers as a return for providing certain service(s)
Biller Category	The category under which billers can be classified, such as: Telecommunications, Electricity, etc...
Biller's Settlement Bank	The bank designated by the biller for the purpose of collecting the biller's net entitlements paid through the system
Billers' Customers	Customers and / or those who deal with the biller and / or payers who have or do not have bank accounts to pay off the due installments through the different payment channels offered by the banks or payment service providers





Con 2 Us
Integration Specs

Bills	Bills and/or claims issued by the biller, which the customer pays off through the different payment channels in favor of the biller
Billing Number	Represents the unique identifier of subscription in a specific service provided by the biller
Business Flow	Specifying business processes in a business process model
Business Process Operations (BPO)	A collection of related, structured activities or tasks that produce a specific service or product

Business Validation	The process of ensuring that the system/product operates according to the agreed-on business rules
CENTRAL BANK	The Central Bank of Jordan
Credit	An entry made in account ledgers to record changes in value resulting from business transactions
CSV	Comma Separated Values
Customer Profile	The customer information at ONEIC-TASDEED
Data Validation	The process of ensuring that the system operates on clean, correct, and valid data
Debit	An entry made in account ledgers to record changes in value resulting from financial transaction(s)
Digital Certificate	An electronic 'passport' that allows all participants to exchange the information securely over the transportation channel using the public key infrastructure (PKI)
Digital Signature	A digital code generated by the private key of the participant to verify its contents and the sender's identity





Con 2 Us
Integration Specs

ONEIC-TASDEED	A system owned by Central Bank of Jordan, and operated by 'MFEP' that aims to work as an electronic payment switch between all Banks/PSPs from a side and Billers from the other side
ONEIC-TASDEED Portal	The web interface of ONEIC-TASDEED in which the system users can manage and configure all administration and operational areas related to ONEIC-TASDEED
Error Code	Enumerated message that correspond to an occurred fault in a specific area of a system
Exception	Is an event, which occurs during the execution of a system
Exist Bill	A bill which is uploaded or added to the system previously

Fees	A fixed sum of money or percentage of the value of the bill and / or claim issued by the biller that shall be paid in return for collecting and/or paying the bill and/or claim and / or in return for any other services provided by the system
Fees on Biller	A Boolean value which have True or False; if True, the fees amount will be charged from biller, and if False, the fees amount will be charged from customer
Flat File	Describes any of various means to encode a database model as a single file
Footer	The part of XML message under the message body that contains data related to the message
FTP	File Transfer Protocol
GUID	A (Global Unique Identifier) key that is generated from participant side or ONEIC-TASDEED side and is sent in each single request to identify the transaction





C o n 2 U s
Integration Specs

Header	The part of XML message above the message body that contains data related to the message
Inactive Participant	The Participant status is Inactive, thus, cannot communicate with ONEIC-TASDEED services
Investigations	The action of investigating why a problem, issue, or conflict has occurred
JOEBPPS	Jordan Electronic Bill Payment and Presentment System which has the name 'ONEIC-TASDEED'
JOEBPPS Number	The JOEBPPS Number is the customer subscription number at ONEIC-TASDEED which is a unique number assigned by ONEIC-TASDEED to each customer. This number is used to identify customers for provision value-added services
Logs	Contains a list of events which logged by a system

MadfooatCom	The company duly authorized by CENTRAL BANK to operate and administer ONEIC-TASDEED system and its infrastructure
MFEP	MadfooatCom For Electronic Payments
National ID Number	The national ID number is a unique number for identifying Jordan Nationals and expatriate residents that is assigned by the Ministry of The Interior. JOEBPPS will use this number as the unique customer identifier
Non-exist Bill	A bill which is not-exist in ONEIC-TASDEED database because it was not uploaded or added to the system previously
Non-repudiation	Non-repudiation refers to a state of affairs where the purported maker of a statement will not be able to successfully challenge the validity of the statement or contract





Con 2 Us
Integration Specs

Official ID Type	Used with its corresponding value to differentiate the customer, e.g. National ID, Passport ID
Offline Operation	The operation will be done in a disconnected state (asynchronous)
One-Off	The bill recurrence is for once
Online Operation	The operation will be done on online mode using web services
Participant	The party directly connecting with the system for the purposes of providing services, such as banks, billers, payment service providers, or any party approved by the CENTRAL BANK
Paying Bank	The Bank that initiates the payment request
Payment Acknowledgement	Message used to alert the 'Paying Bank' and/or 'PSP' and/or 'PSP Settlement Bank' of the details of the payment succeeded
Payment Channels	Self-payment channels available all day long, as well as bank branches and payment service providers, which aim to

	provide various banking services in a convenient and easy manner
Payment Details	The information related to a specific payment transaction
Payments History	All payment transactions that was done on a specific bill over time
Payment Method	The possible payment methods used for payment transaction, e.g. cash, credit card, etc...
Payment Notification	Message used to alert Biller and Biller's Settlement Bank of the funds collected on a given date by Paying Bank(s) or PSP(s)





Con 2 Us
Integration Specs

Payment Service Provider (PSP)	The party approved by the CENTRAL BANK to connect with the system and provide the service through which a "customer" pays the amounts owed thereby to the "biller" through the payment channels provided thereby
Payment Status	The payment status of a specific payment transaction, e.g. New, Sent, Completed
PIN Code	Postal Index Number used for Prepaid Services
PKI	Public Key Infrastructure
Prepaid Service	A service for which credit is paid in advance, prior to using the service
PSP's Settlement Bank	The bank assigned by the payment service provider for the purpose of performing and ensuring the settlement of its obligations
Pull Mode	Bills accessed non-proactively by requesting bill(s) data from the biller, and the Biller billing application must respond with the corresponding bill(s) data.
Push Mode	Bills accessed proactively by uploading bills by the biller, with no request initiated by ONEIC-TASDEED prior to it.

Receiver	The participant who gets or receives the message that has been sent (request/response)
Reconciliation	The process of matching the payment records between ONEIC-TASDEED side and participants side
Reconciliation File	The file that is generated in Reconciliation process that contains all payment transactions of that business day per participant





Con 2 Us
Integration Specs

Recurring	The frequency of generating bills is periodic
Repeatable	The container in the XML file can be repeated within the same message
Request	A message initiated by one which requires a proper response from the another side that the request was sent to
Response	A message initiated by a side that has received a request message as an answer for it
REST	REpresentational State Transfer
RTGS	An electronic central multi-currency final-payment system that processes the transfer and settlement of money between member banks on real time basis through their CENTRAL BANK accounts
Semantic Validation	The validation process done to check whether the data conforms to specified business rules
Sender	The participant that sent the message (request/response)
Service Binding	A set of configurations used to specify the type of transportation, protocol, and traffic encoding for a specific service
Service Type	The type of the biller service (e.g. electricity, internet, etc...)
Settlement	The process of calculating the net transfers and obligations of banks that resulted from clearing the transactions that occurred through the system, then sending them to the RTGS-JO or any clearing or settlement system approved by
	the CENTRAL BANK for recording on the accounts of banks that are involved in the settlement process
Severity	Represents the classification of the occurred error





Con 2 Us
Integration Specs

SFTP	SSH File Transfer Protocol (Secure FTP)
SHA-2	A set of cryptographic hash functions
SOAP	Simple Object Access Protocol is used for exchanging structured information in its implementation
SOAP Element	The element contained within the SOAP message
Standalone Application	It is a web portal that provides various features for participants such as: view the digital files and reports being provided, reconciliation purposes, etc...)
STP	Straight Through Processing is an order processing that does not require any manual intervention and is fully automatic
Sub-Payments	The concept of distributing the bill amount into multiple bank accounts of the biller
Suspense Account	Intermediary Account with off-balance sheet records used usually to reconcile payments or other entries at the end of day with balance sheet accounts
Syntax Validation	The validation process done to check whether the XML message conforms to the XML schema
System Operator	The party duly authorized by CENTRAL BANK to operate and administer the system and its infrastructure, or any other party in lieu thereof
The System	The Electronic Bill Presentment and Payment System (ONEIC-TASDEED)
Thick Consolidation	It is a model of bill information acquisition in which the consolidator holds both summary and detailed level information





Con 2 Us
Integration Specs

Thin Consolidation	It is a model of bill information acquisition in which the consolidator holds only summary level information
TimeStamp	A timestamp is a sequence of characters of time and date identifies when a certain event occurred
Token	A value that is given to a participant to be authorized to send a message across the ONEIC-TASDEED network
Token Validation	Validating the token value sent in a participant message to ensure it is authorized to communicate with ONEIC-TASDEED solution
URL	Uniform Resource Locator is a formatted text string used by web browsers to identify a network resource on the Internet.
Web Service	A web service is a software function provided at a network address over the web or the cloud.
Business Day	Official working hours of the RTGS-JO as announced by the CENTRAL BANK during which financial transactions carried out through the system are settled on the banks' accounts at the CENTRAL BANK.
XML	Extensible Markup Language is a markup language that defines a set of rules for encoding documents.
XPath	XML Path Language is a query language for selecting nodes from an XML document.
CSPD	Civil Status and Passport Department of Jordan.





1. Introduction

1.1 Introduction

The ONEIC-TASDEED Integration Interface is a web service communication module that enables Banks, PSPs, and Billers to send and receive financial requests through the ONEIC-TASDEED network for the purpose of bills presentment and payment.

These requests are verified and processed in ONEIC-TASDEED system, which is responsible for handling all transactions in a secured environment.

ONEIC-TASDEED through its solution provides features for tracing the state of the transactions at all stages.

Moreover, ONEIC-TASDEED is integrated with several solutions for the purpose of reconciliation, reporting, clearing, and monitoring.

1.2 Document Purpose

This document describes ONEIC-TASDEED Integration Web Service and its usages as needed when implementing a connector for participant system to interconnect with ONEIC-TASDEED switch.

1.3 Document Scope

This document applies to **ONEIC-TASDEED System Version_2.0.6.1**.

1.4 Audience

This document is intended for development engineers from the participants' system providers that have to enhance their systems to connect to ONEIC-TASDEED switch.

Knowledge of Web Service concepts and technologies is required.



C o n 2 U s
Integration Specs

1.5 Reference Documents

- [XMLSCHEMA] W3C Recommendation (2 May 2001): "XML Schema Part 2: Data types". Available at: <http://www.w3.org/TR/2001/REC-xmlschema-2>
<http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>.
- [XMLSIG] W3C Recommendation (12 February 2002): "XML-Signature Syntax and Processing". Available at: <http://www.w3.org/TR/2002/REC-xmldsig#>
<http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/>
- [WSI] WS-I Basic Profile Version 1.0: "Final Material". Available at:
<http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html>.
- [WSDL] W3C Note (15 March 2001): "Web Services Description Language (WSDL) 1.1". Available at: <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- [WSS] OASIS Standard 200401 (March 2004): "Web Services Security: SOAP Message Security 1.0(WS-Security 2004)".Available at:
<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>
ISO 4217: "Codes for the representation of currencies and funds".
- [ISO4217]



2. ONEIC-TASDEED Switch

ONEIC-TASDEED solution has the ability to connect different Banks and PSPs with different Billers, and at the same time, the solution integrates with the RTGS and the ACH for settlement.

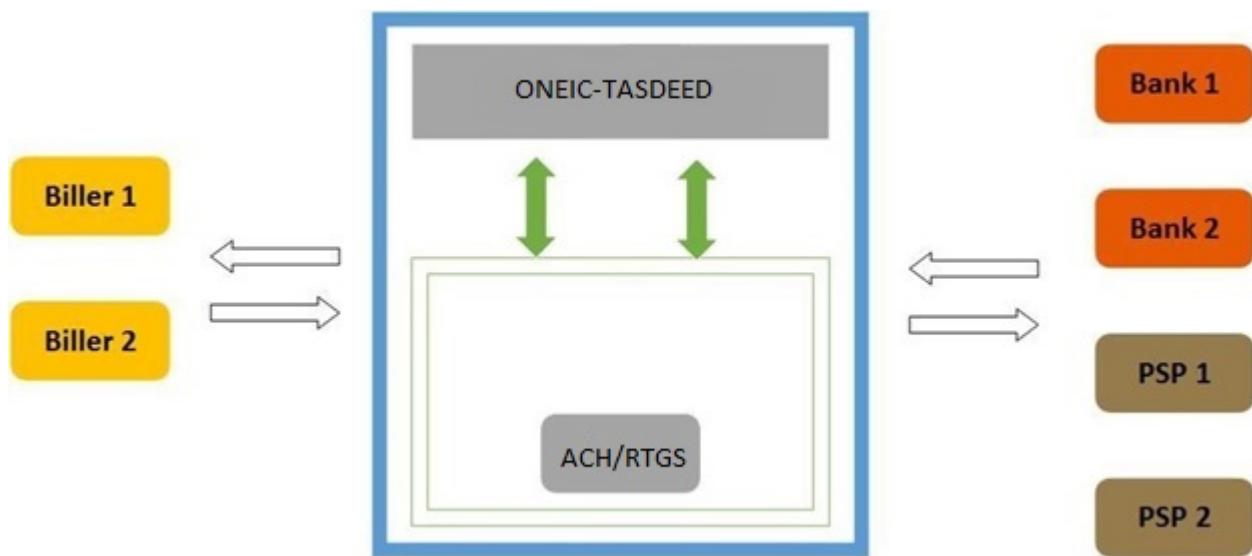


Figure 1 ONEIC-TASDEED Switch



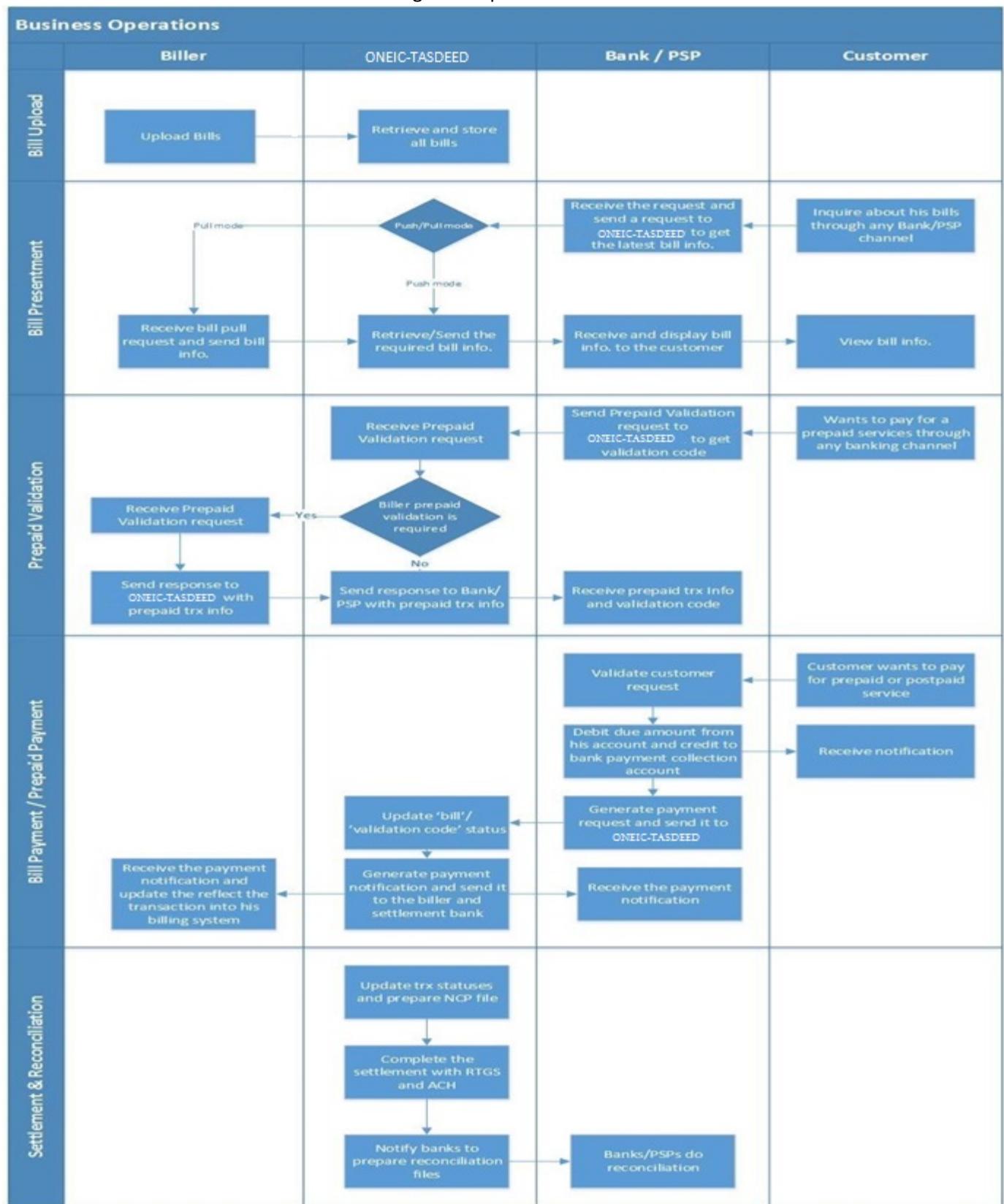
2.1 Business Process Operations (BPOs) of ONEIC-TASDEED

The following workflow shows the main stages that ONEIC-TASDEED consists of:

- Bill Upload Process
- Bill Presentment (Bill Inquiry) Process
- Prepaid Process
- Payment Process
- Settlement and Reconciliation Processes

Note: The solution is capable of supporting different types of payments in addition to handling all payment statuses (New, Sent, and Completed).

Con 2 Us
Integration Specs



2.2 Bill Upload Process

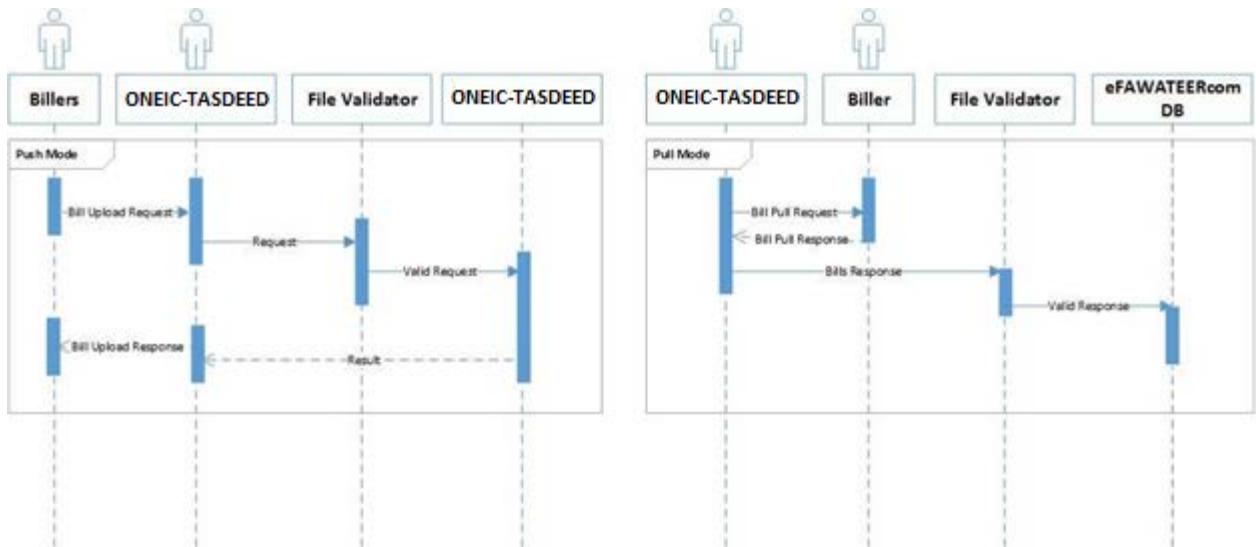


Figure 3 Bill Upload Process

The previous workflow describes in general the bill upload process:

- Billers are required to upload bill summary data to ONEIC-TASDEED on a regular basis using the Bill Upload Process; this process can be:
 - Biller initiated (Push) via Web Service or FTP/SFTP using XML format.
 - ONEIC-TASDEED initiated (Pull) via web service using XML file structure, and can be performed through ONEIC-TASDEED.
- On receiving the uploaded bills, ONEIC-TASDEED performs certain validations on the bills to maintain bills data accuracy. These are:
 - Data Validations.
 - Business Validations.
- If the file/batch has errors/inconsistencies, the system rejects the entire file/batch of records and return it to the biller for reprocessing, and it will mention the rejection reason.

Con 2 Us

Integration Specs

- Each bill on ONEIC-TASDEED database carries a code that shows the status of the bill such as Bill New, or Bill Updated.
- The solution will respond to billers after a successful bill upload operation is performed successfully.

2.3 Bill Presentment (Bill Inquiry) Process

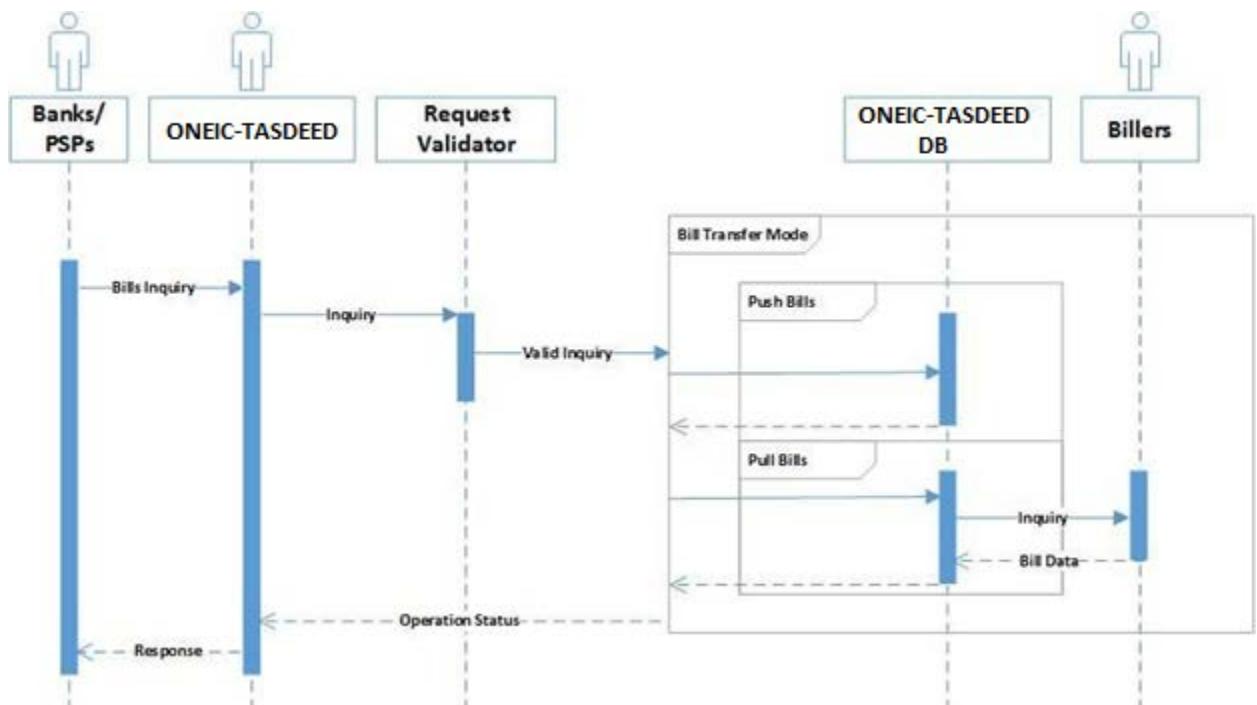


Figure 4 Bill Presentment

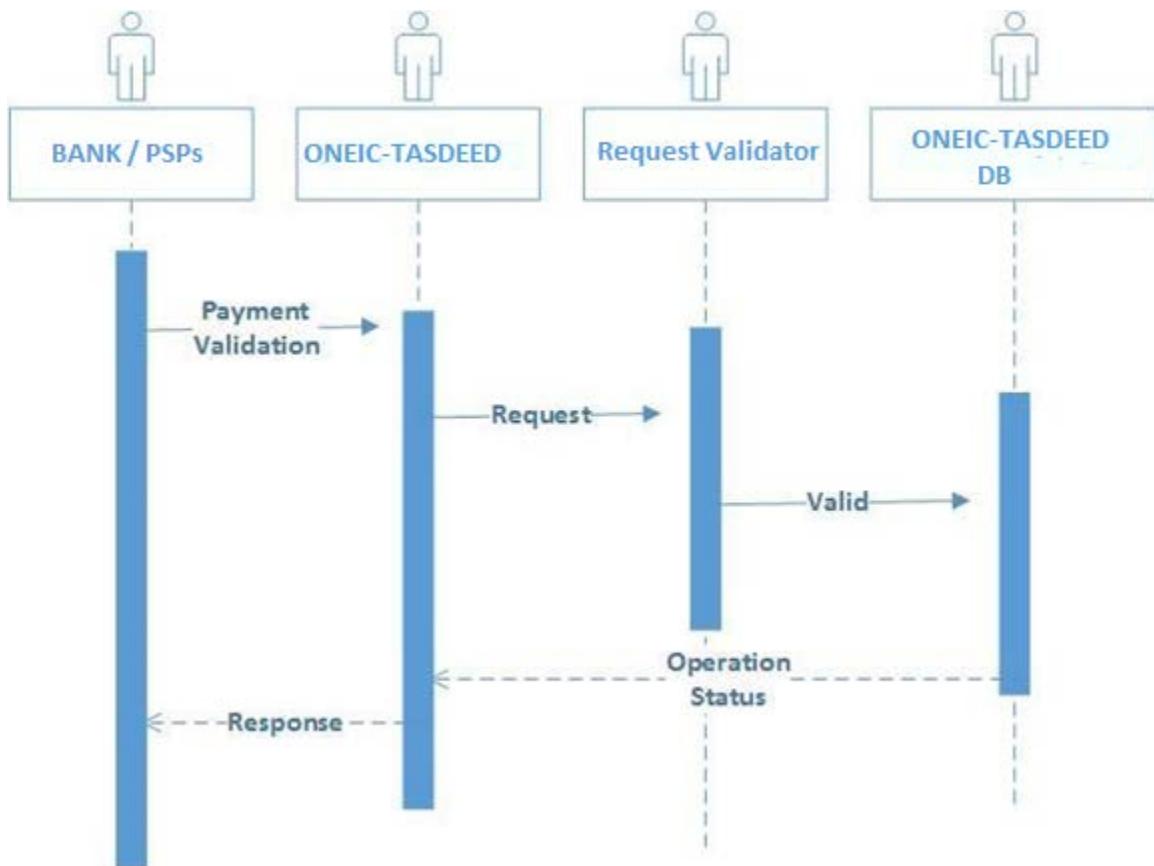
The previous workflow describes in general the bill presentment process:

- Bank/PSP applications may query ONEIC-TASDEED for bill and associated payment data using a bill inquiry message. The query can take the form of a Bill-Specific (single) Query in which the Bank/PSP wishes to view bill data for a specific account or bill number. Conversely, a Customer Profile query permits the Bank/PSP to query on any Customer associated bills (multiple) within the ONEIC-TASDEED system using a variety of parameters.

Con 2 Us
Integration Specs

- The bill inquiry request contains a set of information that entered by the customer such as 'Bill No.' plus a set of information that are provided from the Bank/PSP application such as (Biller Code, Billing Number).
- ONEIC-TASDEED verifies all the business rules (active, Inactive, etc...) to be validated for each request, and based on the verification result, it either accepts or rejects the request.
- The response of bill inquiry may contain one or more records based on the criteria used in the query and might return zero '0' results as well.
- All transactions can occur across a wide range of channels such as ATM, Internet Banking, Bank Teller, Call Center, PSP Channels, etc...

2.5 Payment Process (Postpaid & Prepaid Payments)





C o n 2 U s
Integration Specs
Figure 7 Bill Payment Sequence

The previous workflow describes in general the 'bill payment'/'prepaid payment' process:

- The bill payment and prepaid payment processes permits Banks/PSPs to create new payment records in ONEIC-TASDEED. The process is intended to ensure the customer pays according to the Biller intent, and it involves a validation of Biller's payment rules.
- If the funds are not sufficient, the Bank/PSP should not send a bill payment request to ONEIC-TASDEED.
- A suspense account will be set-up in each 'Paying Bank', 'PSP Settlement Bank'.
- For prepaid payments, Bank/PSP should send the validation code they got in the corresponding prepaid validation response from ONEIC-TASDEED.
- Banks/PSPs must record data about all payments in storage termed as ONEIC-TASDEED Payment Log.
- All transactions can occur across a wide range of channels such as ATM, Internet Banking, Bank Teller, Call Center, PSP Channels, etc...



2.6 Settlement & Reconciliation Processes

- The settlement process should always be ahead of the reconciliation process, where the Banks/PSPs will be given a chance after settlement to reconcile the payments for that day with their own records.
- ONEIC-TASDEED sends net clearing positions files to RTGS (STMTR), one for payments and another one for fees that includes netting of all payments/fees. RTGS will process the 'STMTR' file and send the response 'STMTA' to ONEIC-TASDEED system.
- ONEIC-TASDEED allows banks to reconcile their payment transactions using the standalone application where each Bank / PSP is supposed to upload its data and match with ONEIC-TASDEED data, for more information please refer to 'Settlement and Reconciliation' document.
- For details about these two processes, please refer to 'Settlement and Reconciliation' document.

3. Participants Preparations for ONEIC-TASDEED

- Banks, PSPs, and Billers are supposed to prepare the needed interfaces from their side to interact with ONEIC-TASDEED.
- For all connectivity specifications required by each participant, please refer to the document 'Requirements for Connectivity with ONEIC-TASDEED Gateway'.
- There are certain requirements for the interface that each participant should prepare to communicate successfully with ONEIC-TASDEED for certain processes, such as (Bill Pull, Biller Prepaid Validation, Payment Notifications, Payment Acknowledgement).
- Time out expected from participants in their responses should not exceed a specific period (configurable, currently set to 'one minute').

3.1 Billers Preparations

Billers are supposed to prepare two interfaces that supports the following services:

- Bill Pull (for online inquiry about bills)
- Biller Prepaid Validation (to validate prepaid transactions)
- Biller Payment Notification (for receiving notifications on payments done by Banks/PSPs online)

The interfaces are supposed to receive certain elements and return certain element as well as mentioned in each section separately 'Bill Pull Service', 'Biller Prepaid Validation Service', and 'Biller Payment Notification Service'.

However, the interface that is required to support the 'Bill Pull' and 'Biller Payment Notification' services should be as in the below figure:

```

/// <summary>
/// This Interface is required to be implemented from Billers in order to receive the
/// Bill Pull and payment notification messages from eFawateerCom system.
/// </summary>
[ServiceContract]
public interface IBillerServices
{
    /// <summary>
    /// The BillPull function will be responsible to receive the bill pull message in order to get the latest information of the bill.
    /// </summary>
    /// <param name="guid">This Identifier (Globally Unique Identifier) will be generated by
    /// eFawateerCom in order to reference the requests.</param>
    /// <param name="BillPullRequest">The generated XML message from eFawateerCom
    /// gateway which contains the requested bill information</param>
    /// <param name="username">The username given to eFawateerCom to be used to access Biller interface (Optional)</param>
    /// <param name="password">The password given to eFawateerCom to be used to access Biller interface (Optional)</param>
    /// <returns>XElement variable that contains the response of the payment notification</returns>
    [OperationContract]
    [WebInvoke(Method = "POST", ResponseFormat = WebMessageFormat.Xml,
        RequestFormat = WebMessageFormat.Xml,
        BodyStyle = WebMessageBodyStyle.Bare,
        UriTemplate = "BillPull/?GUID={GUID}&username={username}&password={password}")]
    XElement BillPull(string GUID, XElement BillPullRequest, string username = null, string password = null);

    /// <summary>
    /// The ReceivePaymentNotification function will be responsible to receive the payment notification message.
    /// </summary>
    /// <param name="guid">This Identifier (Globally Unique Identifier) will be generated by
    /// eFawateerCom in order to reference the requests.</param>
    /// <param name="paymentNotification">The generated XML message from eFawateerCom
    /// gateway which contains the notification information</param>
    /// <param name="username">The username given to eFawateerCom to be used to access Biller interface (Optional)</param>
    /// <param name="password">The password given to eFawateerCom to be used to access Biller interface (Optional)</param>
    /// <returns>XElement variable that contains the response of the payment notification</returns>
    [OperationContract]
    [WebInvoke(Method = "POST", ResponseFormat = WebMessageFormat.Xml,
        RequestFormat = WebMessageFormat.Xml,
        BodyStyle = WebMessageBodyStyle.Bare,
        UriTemplate = "ReceivePaymentNotification/?guid={GUID}&username={username}&password={password}")]
    XElement ReceivePaymentNotification(string guid, XElement paymentNotification, string username = null, string password = null);
}

```



Con 2 Us

Integration Specs

Figure 8 Biller Interface for Bill Pull and Biller Payment Notification

Whereas the interface that is required to support the 'Biller Prepaid Validation' service should be as in the below figure:

```
[ServiceContract]
public interface IBillerPrepaid
{
    /// <summary>
    /// The PrepaidValidation function will be responsible to receive the prepaid message in order to get the due balance.
    /// </summary>
    /// <param name="guid">This Identifier (Globally Unique Identifier) will be generated by
    /// eFawateerCom in order to reference the requests.</param>
    /// <param name="billerPrepaidValidationRequest">The generated XML message from eFawateerCom
    /// gateway which contains the prepaid request information</param>
    /// <param name="username">The username given to eFawateerCom to be used to access Bank/PSP interface (Optional)</param>
    /// <param name="password">The password given to eFawateerCom to be used to access Bank/PSP interface (Optional)</param>
    /// <returns>XElement variable that contains the response of the prepaid order</returns>
    /// <returns></returns>
    [OperationContract]
    [WebInvoke(Method = "POST", ResponseFormat = WebMessageFormat.Xml,
    RequestFormat = WebMessageFormat.Xml,
    BodyStyle = WebMessageBodyStyle.Bare,
    UriTemplate = "PrepaidValidation/?GUID={guid}&username={username}&password={password}")]
    XElement PrepaidValidation(string guid, XElement billerPrepaidValidationRequest, string username = null, string password = null);
}
```

Figure 9 Biller Interface for Biller Prepaid Validation

Note: The implementation itself might differ from a technology to another, however, the parameters, data types, and service binding should be similar to the information in the sample above.

3.2 Banks Preparations

Settlement Banks ('Biller Bank' or 'PSP Settlement Bank') are supposed to prepare four interfaces that supports the following services:

- Bank Payment Notification (for receiving notifications on payments done by Banks/PSPs online)
- Payment Acknowledgment (for receiving acknowledgments on successful payments done by the Bank itself or its PSP(s) online)
- System Service Availability (for receiving the current status of the ONEIC-TASDEED system, such as: UP, Down).
- Biller Service Availability (for receiving the current status of the Billers billing system, such as: UP, Down).





Con 2 Us

Integration Specs

The interface is supposed to receive certain elements and return certain elements as well as mentioned in each section separately 'Bank Payment Notification Service', 'Payment Acknowledgment Service', 'System Service Availability', and 'Biller Service Availability'.

However, the interface that is required to support the 'Bank Payment Notification' service should be as in the below figure:

```
/// <summary>
/// This Interface is required to be implemented from Banks and PSPs in order to receive the
/// payment notification message from eFawateerCom system.
/// </summary>
[ServiceContract]
public interface IBankServices
{
    /// <summary>
    /// The ReceivePaymentNotification function will be responsible to receive the payment notification message.
    /// </summary>
    /// <param name="guid">This Identifier (Globally Unique Identifier) will be generated by
    /// eFawateerCom in order to reference the requests.</param>
    /// <param name="paymentNotification">The generated XML message from eFawateerCom
    /// gateway which contains the notification information</param>
    /// <param name="username">The username given to eFawateerCom to be used to access Bank/PSP interface (Optional)</param>
    /// <param name="password">The password given to eFawateerCom to be used to access Bank/PSP interface (Optional)</param>
    /// <returns>XElement variable that contains the response of the payment notification</returns>
    [OperationContract]
    [WebInvoke(Method = "POST", ResponseFormat = WebMessageFormat.Xml,
        RequestFormat = WebMessageFormat.Xml,
        BodyStyle = WebMessageBodyStyle.Bare,
        UriTemplate = "ReceivePaymentNotification/?guid={guid}&username={username}&password={password}")]
    XElement ReceivePaymentNotification(string guid, XElement paymentNotification, string username = null, string password = null);
}
```

Figure 10 Bank Interface for Bank Payment Notification

Whereas the interface that is required to support the 'Payment Acknowledgement' service should be as in the below figure:





Con 2 Us

Integration Specs

```
/// <summary>
/// This Interface is required to be implemented from Banks and PSPs in order to receive the
/// payment acknowledgment message from eFawateerCom system.
/// </summary>
[ServiceContract]
public interface IPaymentAcknowledgment
{
    /// <summary>
    /// The ReceivePaymentAcknowledgment function will be responsible to receive the payment acknowledgment message.
    /// </summary>
    /// <param name="guid">This Identifier (Globally Unique Identifier) will be generated by
    /// eFawateerCom in order to reference the requests.</param>
    /// <param name="paymentAcknowledgment">The generated XML message from eFawateerCom
    /// gateway which contains the acknowledgment information</param>
    /// <param name="username">The username given to eFawateerCom to be used to access Bank/PSP interface (Optional)</param>
    /// <param name="password">The password given to eFawateerCom to be used to access Bank/PSP interface (Optional)</param>
    /// <returns> XElement variable that contains the response of the payment acknowledgment</returns>
    [OperationContract]
    [WebInvoke(Method = "POST", ResponseFormat = WebMessageFormat.Xml,
    RequestFormat = WebMessageFormat.Xml,
    BodyStyle = WebMessageBodyStyle.Bare,
    UriTemplate = "ReceivePaymentAcknowledgment/?guid={guid}&username={username}&password={password}")]
    XElement ReceivePaymentAcknowledgment(string guid, XElement paymentAcknowledgment, string username = null, string password = null);
}
```

Figure 11Bank Interface for Payment Acknowledgment

While the interface that is required to support the 'System Service Availability' service should be as in the below figure:

```
/// <summary>
/// This Interface is required from Banks and PSPs to be implemented in order to receive the
/// status of eFawateerCom system during a specific period.
/// </summary>
[ServiceContract]
public interface ISysServiceAvailability
{
    /// <summary>
    /// The RecvSysSvcAvailabilityStuts function will be responsible to receive the system
    /// availability status from eFawateerCom gateway.
    /// </summary>
    /// <param name="guid">This Identifier (Globally Unique Identifier) will be generated by
    /// eFawateerCom in order to reference the requests.</param>
    /// <param name="availabilityNotificationRequest">The generated XML message from eFawateerCom
    /// gateway which contains the availability information</param>
    /// <returns></returns>
    [OperationContract]
    [WebInvoke(Method = "POST", ResponseFormat = WebMessageFormat.Xml,
    RequestFormat = WebMessageFormat.Xml,
    BodyStyle = WebMessageBodyStyle.Bare,
    UriTemplate = "RecvSysSvcAvailabilityStuts/?guid={guid}")]
    bool RecvSysSvcAvailabilityStuts(string guid, XElement availabilityNotificationRequest);
}
```

Figure 12 System Availability Status Interface

The interface that is required to support the 'Biller Service Availability' service should be as in the below figure:





Con 2 Us

Integration Specs

```
/// <summary>
/// This Interface is required from Banks and PSPs to be implemented in order to receive the current status of the Billers system.
/// </summary>
[ServiceContract]
public interface IBillerServiceAvailability
{
    /// <summary>
    /// The RecvBillerSvcAvailabilityStuts function will be responsible to receive the current status of biller system.
    /// </summary>
    /// <param name="guid">This identifier (Global Unique identifier) will be generated by eFawateerCom in order to
    /// reference the requests</param>
    /// <param name="billerSrvStatus">The generated XML message from eFawateerCom gateway which contains the availability of the Biller system</param>
    /// <returns>true/false</returns>
    [OperationContract]
    [WebInvoke(Method = "POST", ResponseFormat = WebMessageFormat.Xml,
    RequestFormat = WebMessageFormat.Xml,
    BodyStyle = WebMessageBodyStyle.Bare,
    UriTemplate = "RecvBillerSvcAvailabilityStuts/?guid={guid}")]
    bool RecvBillerSvcAvailabilityStuts(string guid, XElement billerSrvStatus);
}
```

Figure 13 Biller Availability Status Interface

Note: The implementation itself might differ from a technology to another, however the parameters, data types, and service binding should be similar to the information in the samples above.

3.3 PSPs Preparations

PSPs are supposed to prepare one interface that supports the following service:

- Payment Acknowledgment (for receiving acknowledgments on its successful payments)
- System Service Availability (for receiving the current status of the ONEIC-TASDEED system, such as: UP, Down).
- Biller Service Availability (for receiving the current status of the Billers billing system, such as: UP, Down).

The interface is supposed to receive certain elements and return certain elements as well as mentioned in the section of 'Payment Acknowledgment Service', 'System Service Availability', and 'Biller Service Availability'.

Whereas the interface that is required to support the 'Payment Acknowledgement' service should be as in the below figure:





Con 2 Us

Integration Specs

```
/// <summary>
/// This Interface is required to be implemented from Banks and PSPs in order to receive the
/// payment acknowledgment message from eFawateerCom system.
/// </summary>
[ServiceContract]
public interface IPaymentAcknowledgment
{
    /// <summary>
    /// The ReceivePaymentAcknowledgment function will be responsible to receive the payment acknowledgment message.
    /// </summary>
    /// <param name="guid">This Identifier (Globally Unique Identifier) will be generated by
    /// eFawateerCom in order to reference the requests.</param>
    /// <param name="paymentAcknowledgment">The generated XML message from eFawateerCom
    /// gateway which contains the acknowledgment information</param>
    /// <param name="username">The username given to eFawateerCom to be used to access Bank/PSP interface (Optional)</param>
    /// <param name="password">The password given to eFawateerCom to be used to access Bank/PSP interface (Optional)</param>
    /// <returns>XElement variable that contains the response of the payment acknowledgment</returns>
    [OperationContract]
    [WebInvoke(Method = "POST", ResponseFormat = WebMessageFormat.Xml,
    RequestFormat = WebMessageFormat.Xml,
    BodyStyle = WebMessageBodyStyle.Bare,
    UriTemplate = "ReceivePaymentAcknowledgment/?guid={guid}&username={username}&password={password}")]
    XElement ReceivePaymentAcknowledgment(string guid, XElement paymentAcknowledgment, string username = null, string password = null);
}
```

Figure 14 PSP Interface for Payment Acknlowedgment

While the interface that is required to support the 'System Service Availability' service should be as in the below figure:

```
/// <summary>
/// This Interface is required from Banks and PSPs to be implemented in order to receive the
/// status of eFawateerCom system during a specific period.
/// </summary>
[ServiceContract]
public interface ISysServiceAvailability
{
    /// <summary>
    /// The RecvSysSvcAvailabilityStuts function will be responsible to receive the system
    /// availability status from eFawateerCom gateway.
    /// </summary>
    /// <param name="guid">This Identifier (Globally Unique Identifier) will be generated by
    /// eFawateerCom in order to reference the requests.</param>
    /// <param name="availabilityNotificationRequest">The generated XML message from eFawateerCom
    /// gateway which contains the availability information</param>
    /// <returns></returns>
    [OperationContract]
    [WebInvoke(Method = "POST", ResponseFormat = WebMessageFormat.Xml,
    RequestFormat = WebMessageFormat.Xml,
    BodyStyle = WebMessageBodyStyle.Bare,
    UriTemplate = "RecvSysSvcAvailabilityStuts/?guid={guid}")]
    bool RecvSysSvcAvailabilityStuts(string guid, XElement availabilityNotificationRequest);
}
```

Figure 15 PSP Interface for System Availability Status Interface

The interface that is required to support the 'Biller Service Availability' service should be as in the below figure:





Con 2 Us

Integration Specs

```
/// <summary>
/// This Interface is required from Banks and PSPs to be implemented in order to receive the current status of the Billers system.
/// </summary>
[ServiceContract]
public interface IBillerServiceAvailability
{
    /// <summary>
    /// The RecvBillerSvcAvailabilityStuts function will be responsible to receive the current status of biller system.
    /// </summary>
    /// <param name="guid">This identifier (Global Unique identifier) will be generated by eFawateerCom in order to
    /// reference the requests</param>
    /// <param name="billerSrvStatus">The generated XML message from eFawateerCom gateway which contains the availability of the Biller system</param>
    /// <returns>true/false</returns>
    [OperationContract]
    [WebInvoke(Method = "POST", ResponseFormat = WebMessageFormat.Xml,
    RequestFormat = WebMessageFormat.Xml,
    BodyStyle = WebMessageBodyStyle.Bare,
    UriTemplate = "RecvBillerSvcAvailabilityStuts/{guid}")]
    bool RecvBillerSvcAvailabilityStuts(string guid, XElement billerSrvStatus);
}
```

Figure 16 PSP Interface for Biller Availability Status Interface

Note: The implementation itself might differ from a technology to another, however the parameters, data types, and service binding should be similar to the information in the samples above.





4. Integration Process

The following sections summarize integration process and the design concept of ONEIC-TASDEED and provide an overview on ONEIC-TASDEED Schema:

4.1 Customers Unique Identification and Registration

- 4.1.1 ONEIC-TASDEED will not 'own' customers and will not store confidential customers information.
- 4.1.2 The combination (ID, ID Type, Nationality) will be initially used as the key to uniquely identify a customer, and once the customer is registered at ONEIC-TASDEED, ONEIC-TASDEED will assign a unique number for him 'JOEBPPS No'.
- 4.1.3 The Banks/PSPs should inquire ONEIC-TASDEED to check if the customer was registered previously or not, In case the customer is registered on ONEIC-TASDEED and not in Banks/PSPs system; the Banks/PSPs will create a new record in its system based on the returned information from ONEIC-TASDEED without requesting ONEIC-TASDEED to create a new profile for this customer.
- 4.1.4 The Banks/PSPs should inquire ONEIC-TASDEED to check if the customer was registered previously or not, In case the customer is not registered on ONEIC-TASDEED; Banks/PSPs should request ONEIC-TASDEED to create a new profile for this customer and create a new record in its (Banks/PSPs) system based on the information returned from ONEIC-TASDEED.
- 4.1.5 ONEIC-TASDEED will validate the legitimacy of customer information through inquiring the CSPD DB. ONEIC-TASDEED will reject the creation of any profiles with incorrect information (ID, ID Type, Nationality).
- 4.1.6 The Banks/PSPs should send Customer ID (ID, ID Type, Nationality) or ('JOEBPPS No') with each request (Bill Presentment or Payment).
- 4.1.7 Customers do not need to register separately each time in order to use the service; ONEIC-TASDEED will build up a 'payments history' for each customer, linking his ID ('JOEBPPS No') with his billings over time.

4.2 Bill Information Acquisition

- 4.2.1 ONEIC-TASDEED will support both consolidation models (Thin and Thick) by design, however, it relies on the biller service types definition in ONEIC-TASDEED Portal.

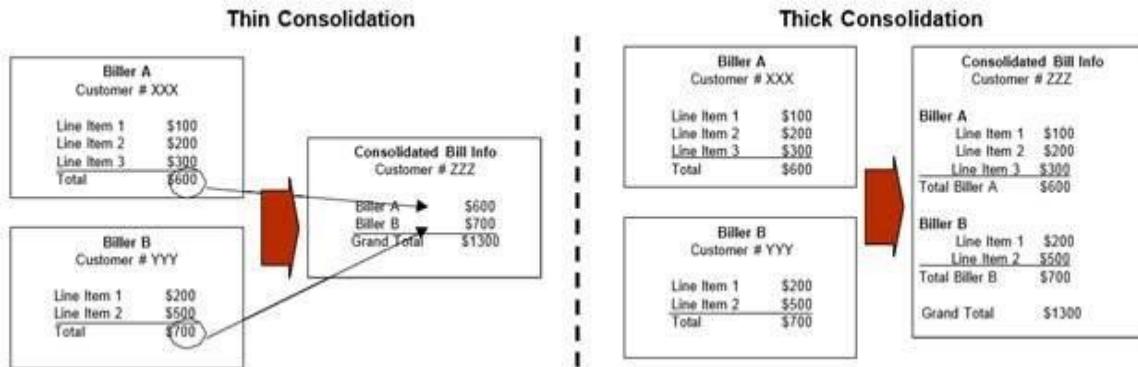


Figure 17 Consolidation Sample

Thin Consolidation:

- Customer must contact respective billers for questions on detailed line items.
- Consolidator holds only summary level information.

Thick Consolidation:

- Consolidator holds both summary and detailed level information.
- Consolidator can answer summary as well as detailed level queries.

- 4.2.2 As part of the biller definition in the biller management module in ONEIC-TASDEED Portal, the biller can be given a consolidation value (Thin or Thick) per service type as shown below:



Con 2 Us Integration Specs

Biller Services								
Service Name	Payment Type	Thick	Fees On Biller	Billing Number Required	Biller Prepaid Validation Required	Min Due Amount	Max Due Amount	
HSPA	Post-Paid	False	True	True	False	1.000	350.000	
Fixed Line	Post-Paid	False	True	True	False	1.000	500.000	
Internet	Post-Paid	False	True	True	False	1.000	1500.000	
Mobile	Pre-Paid	False	False	True	True	-	-	

Figure 18 Consolidation Types Sample

- 4.2.3 ONEIC-TASDEED will support data uploads from billers in both 'Offline' and 'Real Time' modes using XML format.
- 4.2.4 'Thick' model will not be used in the first phase of ONEIC-TASDEED project.

4.3 Bill Inquiry and Bill Presentment

- 4.3.1 ONEIC-TASDEED will provide customers the flexibility to access bill information through Banks/PSPs channels using a variety of identifiers.
- 4.3.2 Bill information can be accessed in both 'push' and 'pull' modes - i.e. all bills accessed proactively/non-proactively in sequence for a specific customer, or specific bills by bill number/billing number.
ONEIC-TASDEED will support access to bill data through all Bank/PSP channels from the outset (i.e. Teller, ATM, Online Banking, Phone Banking, etc...).
- 4.3.3 ONEIC-TASDEED added a new flag for Banks/PSPs 'BillInquiryRefNoRequired' to be able to connect each bill inquiry with bill payment transaction.
- 4.3.4 ONEIC-TASDEED will send the unique reference number in bill inquiry response and Bank/PSP should re-send it in the corresponding bill payment request.

4.4 Prepaid Transactions

- 4.4.1 ONEIC-TASDEED will provide customers the flexibility to pay for prepaid services as well as for postpaid services through Banks/PSPs channels using a variety of identifiers.
- 4.4.2 ONEIC-TASDEED will not process any prepaid payment message without having a





Con 2 Us Integration Specs

validation request for it from Bank/PSP side (prepaid validation message) and there is a unique code for each prepaid transaction used throughout its cycle called 'Validation Code'.

4.5 Payments, Payment Notifications and Acknowledgment

- 4.5.1 Billers will have the ability to provide services with the proper payment mode (postpaid or prepaid) for each of their services for their customers through ONEIC-TASDEED.
- 4.5.2 Billers will have flexibility to select the appropriate payment preferences for each bill (such as: Bill Expiry Date, Bill Close Date, and Exact or Partial/Over Payments).
- 4.5.3 ONEIC-TASDEED will send 'Payment Notifications' to 'Biller' and 'Biller Settlement Bank', and will send 'Payment Acknowledgements' to the 'Paying Bank', 'PSP', and 'PSP Settlement Bank' for each payment that was successfully processed, in both real-time mode and offline mode.

4.6 Advance Payments

- 4.6.1 ONEIC-TASDEED system will make it easier for customers to pay their bills in advance (bills with due amount= '0' or less or more); this feature enables customers to send payments for the billers in advance; even if there is no due amounts on their bills, and/or can pay amounts more than their due amounts based on the billers' preferences.
- 4.6.2 Bill inquiry service will return bills with advance payments feature and allow Banks/PSPs to send it payment requests using the element 'AllowPart'= true which is already used for partial payments as well.

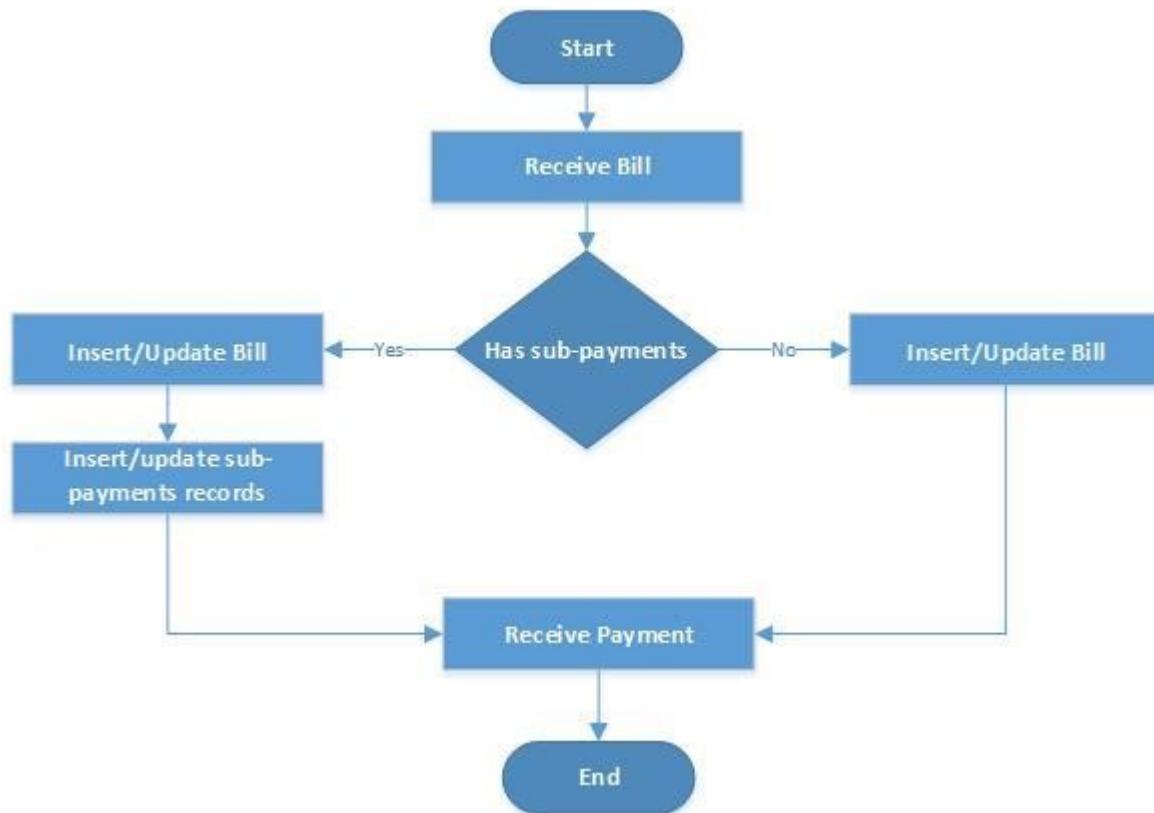
4.7 Offline Mode

- 4.7.1 ONEIC-TASDEED support Offline mode through FTP/SFTP server by defining a set of credentials for each participant (Username, Password, Certificate, File Location, etc...) to store all the inward and outward files.
- 4.7.2 Any Participant can use this mode per its preferences for certain operations as can be found in 'Operations Types' section such as: Bill Upload (Bill Batch), Payment Notification, and Payment Acknowledgment. Also, it can be used for Reconciliation Files, and Settlement Reports.



4.8 Sub-Payments Support

- 4.8.1 ONEIC-TASDEED supports the sub-payments as a part of the Business Process Operations (BPOs).
- 4.8.2 Payment notification message will contain the sub-payments received in the original bill with the 'Settlement Bank Code', 'Amount', and 'Account Number'.
- 4.8.3 ONEIC-TASDEED added a feature to send sub-payments to bank accounts other than the original settlement bank defined for a biller using 'Allow multiple settlement banks' which does not require any change on connectivity level or on communication level.
- 4.8.4 ONEIC-TASDEED will send payment notification messages for the settlement banks listed in each bill sub-payments.
- 4.8.5 ONEIC-TASDEED will handle the sub-payments in bills as shown in the following diagram:





C o n 2 U s
Integration Specs
Figure 19 Sub-Payments Handling

- 4.8.6 You can find the business rules related to Sub-Payments in the sections 'Business Rules - Bill Upload, Bill Pull, and Payment Notifications'.

4.9 Maximum Number of Records per Message

- 4.9.1 The system is designed to process multiple records within the same message of online operation (bill upload and bill payment), and the current maximum number of records per message' is set to (3 records), however, this number will be modified in the future.
- 4.9.2 Currently, for offline bill batch message, the maximum number of records per message is set to (10000 records).

4.10 Data Types

- 4.10.1 There is a variety of elements used by ONEIC-TASDEED through its gateway messages, and each element has a specific data type that its value should match and validated accordingly, as the following:

Data Type	Description	Example (Element Name)	Spaces Allowed for Value (Start, Inside, End)
Boolean	Stores a logic value (true / false)	AllowPart	Yes, No, Yes
Closed Enum	Stores a predefined values that are different from each other, and that can be compared and assigned, and no new values can be added for that enumerator	PaymentType	No, No, No
Container	Stores multiple elements as child elements	AcctInfo	Yes, Yes, Yes





Con 2 Us
Integration Specs

Date	Stores year, month, and day values, in the form of: 'yyyy-mm-dd'	STMTDate	No, No, No
DateTime	Stores year, month, day, hour, minute, and second values, in the form of: 'yyyy-mm-ddThh:mm:ss'	TmStp	Yes, No, Yes
Decimal	Store on-repeating fractional numbers, to preserve exact precision	PaidAmt	Yes, No, Yes
Email	Stores email addresses for different entities	Email	No, No, No
GUID	Stores a unique random reference number displayed separated by hyphens (Global Unique Identifier)	GUID	No, No, No
IBAN	A standard numbering system developed to identify Bank accounts from around the world	AcctNo	No, No, No
Integer	Stores a whole number (not a fraction) that can be positive, negative, or zero	JOEBPPSNo	Yes, No, Yes
Opened Enum	Stores a predefined values that are different from each other, and that can be compared and assigned, and new values can be added and defined for that enumerator	ServiceType	No, No, No
Phone	Stores phone numbers information and can have special chars (e.g. '+')	Phone	No, No, No





Con 2 Us
Integration Specs

String	Stores a set of characters that contains characters, digits, underscores, or periods, and must begin with either a digit or letter	Address	No, Yes, No
---------------	--	---------	-------------

4.11 Services Elements Specs.

4.11.1 ONEIC-TASDEED defined a variety of specifications per each element used in its gateway messages, and the values sent for each element should match these specs., as the following:

Element Name	Field Description	Data Type	Field Length	Case-Sensitive	Support Arabic
AccessChannel	Represents the payment channel type that used to collect payments	Opened Enum	Up to 15 char	No	No
AcctNo	Represents the account number that is supposed to be used for sending the sub payment amount to billers	String	30 char	No	No
Acct	A group of xml elements contains a customer accounts information in its sub nodes (Children)	Container	-	-	-
AcctInfo	A group of xml elements contains a billing information in its sub nodes (Children)	Container	-	-	-
Accts	A group of xml elements contains a set of customers' accounts information in its sub nodes (Children)	Container	-	-	-





Con 2 Us
Integration Specs

Address	Represents the customer address	String	Up to 100 char	No	Yes
AllowPart	Represents the modality to pay the bill partially or exact	Boolean	Up to 5 char	Yes	No
Amount	Represents the amount of that particular sub payment	Decimal	Up to (10,3)	-	-
ArDesc	Used for prepaid cats and service types, billings	String	Up to 50 char	No	Yes

ArImg	Represents the URL of the Arabic image for its parent element	String	Up to 200 char	No	Yes
ArLogo	Represents the URL of the Arabic logo for its parent element	String	Up to 200 char	No	Yes
ArMsg	Represents a customized Arabic message (e.g. marketing message) for its parent element	String	Up to 200 char	No	Yes
ArName	Represents the Arabic name for its parent element	String	Up to 50 char	No	Yes
ArShortDesc	Represents the Arabic short description for its parent element	String	Up to 20 char	No	Yes
ArShortName	Represents the Arabic short name for its parent element	String	Up to 20 char	No	Yes
BankCode	Represents the unique code of the Bank	Integer	Up to 3 digits	-	-





Con 2 Us
Integration Specs

BankTrxID	Represents the unique transaction number which is generated for banks/PSP in collecting payments	String	Up to 50 char	No	No
BillerCategory	A group of xml elements contains the biller category information in its sub nodes (Children)	Container	-	-	-
BillerCode	Represents the unique code of the Biller	Integer	Up to 3 digits	-	-
BillerInfo	A group of xml elements contains the biller basic information in its sub nodes (Children)	Container	-	-	-
BillerMarketing	A group of xml elements contains the biller marketing	Container	-	-	-

	information in its sub nodes (Children)				
BillerName	A group of xml elements contains the biller name details in its sub nodes (Children)	Container	-	-	-
BillerRec	A group of xml elements contains the complete information for a specific biller in its sub nodes (Children)	Container	-	-	-
BillerService	A group of xml elements contains the complete information of a specific service related to a biller in its sub nodes (Children)	Container	-	-	-





Con 2 Us
Integration Specs

BillerServices	A group of xml elements contains the complete services related to a biller in its sub nodes (Children)	Container	-	-	-
BillersRec	A group of xml elements contains all billers information in its sub nodes (Children)	Container	-	-	-
BillingInfo	A group of xml elements contains the billing information of a specific service related to a biller in its sub nodes (Children)	Container	-	-	-
BillingNo	Represents the unique identifier of subscription in a specific service provided by the biller	String	Up to 50 char	No	Yes
BillingNoRequired	Represents the need of sending Billing Number for a specific Prepaid Service Type	Boolean	Up to 5 char	Yes	No
BillingRec	A group of XML elements that contain the billing account details in its sub nodes (children)	Container	-	-	-

BillingsRec	A group of XML elements that contain the billings details in its sub nodes (children)	Container	-	-	-
BillingStatus	Represents the status of the billing account	Closed Enum.	Up to 15 char	No	No
BillNo	Represents the bill number for a particular billing number	String	Up to 50 char	No	Yes
BillRec	A group of xml elements contains a bill information in its sub nodes (Children)	Container	-	-	-





Con 2 Us
Integration Specs

BillsRec	A group of xml elements contains a set of bills in its sub nodes (Children)	Container	-	-	-
BillStatus	Represents the status of the uploaded bill	Closed Enum.	Up to 20 char	Yes	No
BillType	Represents the type of the uploaded bill	Closed Enum.	Up to 13 char	Yes	No
Category	Represents the category defined for that biller	Opened Enum.	4 char	No	No
CloseDate	Represents the date when the bill is no longer available for payment or inquiry over ONEIC-TASDEED network	DateTime	19 char	Yes	No
Code	Represents the unique code value for its parent element	Integer	Up to 4 digits	-	-
ContainsPrepaidCats	Represents whether a Prepaid Service has Prepaid Categories or not	Boolean	Up to 5 char	Yes	No
Currency	Represents the type of the used currency in payment process	Opened Enum	Up to 5 char	Yes	No
CustId	A group of xml elements contains a customer profile information in its sub nodes (Children)	Container	-	-	-
CustomField	A group of xml elements contains info about the custom field in its sub nodes (Children)	Container	-	-	-
DueAmount	Represents the due amount of the uploaded bill	Decimal	Up to (10,3)	-	-





Con 2 Us
Integration Specs

DueDate	Represents the date when the bill is being required to pay	DateTime	19 char	Yes	No
Email	Represents the email address of the biller/customer	Email	Up to 50 char	No	No
EndDt	Represents the value entered into bill inquiry message to view all bills before this date considering the Bill Issue Date	DateTime	19 char	Yes	No
EnDesc	Used for prepaid cats and service types, billings	String	Up to 50 char	No	No
EnImg	Represents the URL of the English image for its parent element	String	Up to 200 char	No	No
EnLogo	Represents the URL of the English logo for its parent element	String	Up to 200 char	No	No
EnMsg	Represents a customized English message (e.g. marketing message) for its parent element	String	Up to 200 char	No	No
EnName	Represents the English name for its parent element	String	Up to 50 char	No	No
EnShortDesc	Represents the English short description for its parent element	String	Up to 20 char	No	No
EnShortName	Represents the English short name for its parent element	String	Up to 20 char	No	No
ErrorCode	An error code that represents a known system error	Opened Enum.	Up to 3 char		No
ErrorDesc	Represents the description of an error	Opened Enum.	Up to 100 char	No	No





Con 2 Us
Integration Specs

ExpiryDate	For Tokens, represents the expiry date of the token, and for bills, represents the date when the bill is no longer available for payment over ONEIC-TASDEED but still Inquirable	DateTime	19 char	Yes	No
Extra	A group of xml elements contains extra fields and its details in its sub nodes (Children)	Container	-	-	-
FeesAmt	Represents the Total Fees Amount on the payment	Decimal	Up to (10,3)	-	-
FeesOnBiller	Represents whether the fees is on biller or on customer	Boolean	Up to 5 char	Yes	No
FieldID	An ID given to any custom field that is needed to be used in the future	String	Up to 250 char	No	No
FieldName	The custom Element Name or description	String	Up to 250 char	No	No
GUID	Represents a Global Unique Identifier key generated by the Participant or ONEIC-TASDEED and used to identify the transaction, this ID contains 32 digits separated by hyphens as the following 00000000-00000000-0000-000000000000	GUID	36 char	No	No
Id	The identification of the customer based on the chosen value from the IdType field	String	Up to 20 char	No	Yes
IdType	Represents a unique identifier type (The official number used to identify a customer)	Closed Enum.	3 char	Yes	No





Con 2 Us
Integration Specs

IncPaidBills	Determines whether paid bills are to be included in the inquiry response	Boolean	Up to 5 char	Yes	No
IncPayments	Determines whether payment history is to be included in the inquiry response	Boolean	Up to 5 char	Yes	No
InqRefNo	Represents the Inquiry Reference Number which connect the bill inquiry with the corresponding bill payment transaction	Integer	Up to 20 digits	-	-
IntegrationType	Represents the Integration Type for a biller (Online, Offline, etc...)	Opened Enum.	Up to 20 char	No	No
IssueDate	Represents the date when the bill was issued by the biller	DateTime	19 char	Yes	No
JOEBPPSNo	Represents the unique identifier of the customer profile in ONEIC-TASDEED	Integer	Up to 10 digits	-	-
JOEBPPSTrx	Represents the ONEIC-TASDEED transaction number that used as a reference number between all Participants and ONEIC-TASDEED	Integer	Up to 20 digits	-	-
Lower	Represents the lowest possible value to pay	Decimal	Up to (10,3)	-	-
MFEP	Represents the root of the message	Container	-	-	-
MsgBody	Represents the main transaction info.	Container	-	-	-





Con 2 Us
Integration Specs

MsgFooter	Represents the security elements and any extra info.	Container	-	-	-
MsgHeader	Represents the transmit info. and transaction timestamp	Container	-	-	-

MsgLabel	Represents a label or note about a transaction by the biller	String	Up to 500 char	No	Yes
Name	Represents the customer name	String	Up to 50 char	No	Yes
Nation	Represents the nationality of the customer	Closed Enum.	Up to 2 char	Yes	No
NickName	Represents the Nike name of the customer billing.	String	Up to 50 char	No	Yes
OpenDate	Represents the date when the bill is available for payment over ONEIC-TASDEED network	DateTime	19 char	Yes	No
PaidAmt	Represents the collected payment amount	Decimal	Up to (10,3)	-	-
ParentTagXPath	Parent Tag XPath where the custom field is found inside, it helps to access the needed value easily	String	Up to 250 char	No	No
ParFees	Represents the fees of the participant	Decimal	Up to (10,3)	-	-
ParTrxID	Represents the unique transaction number which is generated on collecting the payment by the participant	String	Up to 50 char	No	No





Con 2 Us
Integration Specs

PayerInfo	A group of xml elements contains details about the payer (such as: Name, Address, etc..) in its sub nodes (Children)	Container	-	-	-
PaymentMethod	Represents the method of the payment via Bank/PSP	Opened Enum	Up to 15 char	Yes	No
PaymentType	Represents the type of the payment (Postpaid, Prepaid)	Closed Enum	15 char	Yes	No

Phone	Represents the customer/participant phone number	Phone	(10 - 20) char	No	No
PmtConst	A group of xml elements contains a bill payment constraints per bill in its sub nodes (Children)	Container	-	-	-
PmtGuid	Represents the GUID used in the corresponding payment transaction	GUID	36 char	No	No
PmtSrc	Represents the different source of the payment	Opened Enum.	Up to 15 char	Yes	No
PmtStatus	Represents the status of the collected payment	Closed Enum.	Up to 15 char	Yes	No
PrepaidCat	Represents the prepaid category related to a specific prepaid service of a biller	Opened Enum.	Up to 20	No	No
PrepaidCategories	A group of xml elements contains all prepaid categories related to a specific prepaid service for a biller in its sub nodes (Children)	Container	-	-	-





Con 2 Us
Integration Specs

PrepaidCatInfo	A group of xml elements contains information for a specific prepaid category related to a prepaid service in its sub nodes (Children)	Container	-	-	-
ProcessDate	Represents the payment transaction date	DateTime	19 char	Yes	No
PSPCode	Represents the unique code of the PSP	Integer	Up to 3 digits	-	-
RcvCode	Represents the Participant code or ONEIC-TASDEED code who is receiving the message	Integer	Up to 3 digits	-	-

RecCount	Represents the retrieved records count	Integer	Up to 5 digits	-	-
ReqTyp	Request Type, represents the type of the operation that is being executed	Opened Enum.	Up to 15 char	Yes	No
ResTyp	Represents the type of the operation that is being executed	Opened Enum.	Up to 15 char	Yes	No
Result	A group of xml elements contains the result of the message processing in its sub nodes (Children)	Container	-	-	-
SdrCode	Represents the participant code or ONEIC-TASDEED code who is receiving the message	Integer	Up to 3 digits	-	-
Security	A group of xml elements contains security details in its sub nodes (Children)	Container	-	-	-





Con 2 Us
Integration Specs

Seq	Sequence number, not used currently	Integer	Up to 10 digits	-	-
Sequence	A group of xml elements contains session and sequence in its sub nodes (Children)	Container	-	-	-
ServiceInfo	A group of xml elements contains the basic information related to a specific service for a biller in its sub nodes (Children)	Container	-	-	-
ServiceMarketing	A group of xml elements contains the marketing information related to a specific service for a biller in its sub nodes (Children)	Container	-	-	-
ServiceType	Represents the service type of the biller	Opened Enum.	Up to 25 char	No	No

Sess	Session Number, not used currently	Integer	Up to 10 digits	-	-
SetBnkCode	Represents the code of the 'Settlement Bank'	Integer	Up to 3 digits	-	-
Severity	Represents the severity of the problem that occurred	Closed Enum.	Up to 7 char	Yes	No
Signature	Represents the generated signature using the certificate that given to each Participant for message body	String	Up to 4000 char	No	No
StartDt	Represents the value that will be entered into bill inquiry message to view all bills after this date considering the Bill Issue Date	DateTime	19 char	Yes	No





Con 2 Us
Integration Specs

StmtBankCode	The code of the 'settlement bank' for that biller	Integer	Up to 3 digits	-	-
STMTDate	The generated date by ONEIC-TASDEED which represent when the payment will be settled	Date	10 char	Yes	No
SubPmt	A group of xml elements contains a sub payment information in its sub nodes (Children)	Container	-	-	-
SubPmts	A group of xml elements contains a bill break down (Sub Payments) in its sub nodes (Children)	Container	-	-	-
SystemMarketing	A group of xml elements contains the marketing information related to the system 'ONEIC-TASDEED' in its sub nodes (Children)	Container	-	-	-
TmStp	Time Stamp: When the request/response was originally generated by the participant or ONEIC-TASDEED	DateTime	19 char	Yes	No
TokenConf	A group of xml elements contains a set of Token configurations in its sub nodes (Children)	Container	-	-	-
TokenKey	Represents the generated token value after calculations	String	Up to 67 char	Yes	No
Transactions	A group of xml elements contains details about the transactions in its sub nodes (Children)	Container	-	-	-





Con 2 Us
Integration Specs

TrsInf	A group of xml elements contains transmit details in its sub nodes (Children)	Container	-	-	-
Type	Represents the type value for its parent element	String	Up to 50 char	No	No
Upper	Represents the highest possible value to pay the bill	Decimal	Up to (10,3)	-	-
ValidationCode	Represents the unique code of the Prepaid transaction used to validate it throughout its cycle	String	Up to 50 char	No	No
Value	The value that is given to the custom field node	String	Up to 250 char	No	Yes
Website	Represents the website address	String	Up to 50 char	No	No

4.12 Request/Response GUID

4.12.1 Every single transaction that is received or sent by ONEIC-TASDEED has a unique GUID that identifies this transaction, and the same GUID will be used in the request and the response for the same transaction.

4.13 Service Binding

4.13.1 ONEIC-TASDEED provides a set of service bindings for all participants to specify the transport, encoding, and protocol details required to communicate with each other, which are:

- BasicHttpBinding (SOAP)
- BasicHttpBinding with security mode(SOAP)





Con 2 Us

Integration Specs

- WSHttpBinding (SOAP)
- WSHttpBinding with security mode (SOAP)
- WebHttpBinding (REST)
- WebHttpBinding with security mode (REST)

4.14 Clocks Synchronization

- 4.14.1 In order to get accurate information about transactions exact date/time, all participant are required to synchronize their server clocks with ONEIC-TASDEED NTP server.
- 4.14.2 Details about clock synchronization will be part of integration testing and will be provided by ONEIC-TASDEED.

4.15 Session and Sequence

- 4.15.1 Participants should consider the session and sequence - as part of messages been exchanged- in their design, however, this business will not be available at the first stage, and a separate document will be shared to explain how Participants are to implement it in the future. For more details, please refer to section 'Header and Footer'.

4.16 Integration Approaches

4.16.1 SOAP Approach:

- 4.16.1.1 ONEIC-TASDEED Switch is designed to exchange messages using SOAP (Simple Object Access Protocol) which is used for exchanging structured information in its implementation.
- 4.16.1.2 SOAP is based on XML as the messages format, and relies on the HTTP protocols for transmission.

4.16.2 Soap Elements:

A SOAP message is an ordinary XML document containing the following elements:





Con 2 Us Integration Specs

Element	Description	Required
Envelope	Identifies the XML document as a SOAP message	Yes
Header	Contains header information	No
Body	Contains request and response information	Yes
Fault	Provides information about errors that occurred while processing the message	No

SOAP Sample:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:r="http://schemas.xmlsoap.org/ws/2005/02/rm"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <r:Sequence s:mustUnderstand="1">
      <r:Identifier>urn:uuid:25c00033-ea23-4b32-946f-eeac0f2a8f0c</r:Identifier>
      <r:MessageNumber>1</r:MessageNumber>
    </r:Sequence>
    <a:Action s:mustUnderstand="1">http://tempuri.org/ITokenService/Authenticate</a:Action>
    <a:MessageID>urn:uuid:ddc89f2e-7d2e-4ccc-9f61-844e7a8999a1</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">http://172.20.0.22:7011	TokenName.svc</a:To>
  </s:Header>
  <s:Body>
    <Authenticate xmlns="http://tempuri.org/">
      <guid>924e7261-c072-4a1b-8f2d-b0f28b2bb235</guid>
      <customerCode>27</customerCode>
      <password>18DCA598D20093680EF7800DE35E0B87</password>
    </Authenticate>
  </s:Body>
</s:Envelope>
```

Figure 20 SOAP Sample

For more information, please visit: <http://en.wikipedia.org/wiki/SOAP>

4.16.3 REST Approach

4.16.3.1 ONEIC-TASDEED added a new approach to exchange messages using RESTful Web Services. RESTful Web Services are REST architecture based web services, it is





Con 2 Us

Integration Specs

light weight, highly scalable and maintainable, and very commonly used to create APIs for web based applications.

- 4.16.3.2 RESTful web services make use of HTTP protocol as a medium of communication between client and server. ONEIC-TASDEED will use the http verb (POST) to transfer the XML messages with participants in addition to the already existing approach of SOAP.
- 4.16.3.3 All ONEIC-TASDEED gateway services will be published as a REST service on a different URI (case insensitive) to facilitate the integration operations.

4.17 Calling ONEIC-TASDEED Switch Services

- 4.17.1 Each service is published on a specific URL, where the Bank, PSP, or the Biller will be requested to use a certain URL to perform a certain operation, and only a given URL will be accessible for each participant.
- 4.17.2 Each Participant should get a Token in order to initiate any operation. To call the Authenticate web service that is responsible for performing this operation, the Participant have to send the following parameters:
 - GUID
 - Customer Code
 - Password
- 4.17.3 If ONEIC-TASDEED is the caller for the operation, ONEIC-TASDEED will send only (GUID and XML) in all its requests, and only XML in all its responses.
- 4.17.4 After the participant gets a token, the participant is ready to perform any operation (Inquire about bills, pay a bill, etc...), the participant has to call the correspondent web service that is responsible for performing this operation, and this requires the caller (Participant) to send the following parameters:
 - GUID





C o n 2 U s
Integration Specs

- Token Key
- XML Request

All services in ONEIC-TASDEED switch (except Billers List service, Authenticate (Token)) will require these three parameters to be present in request as in the following diagram:

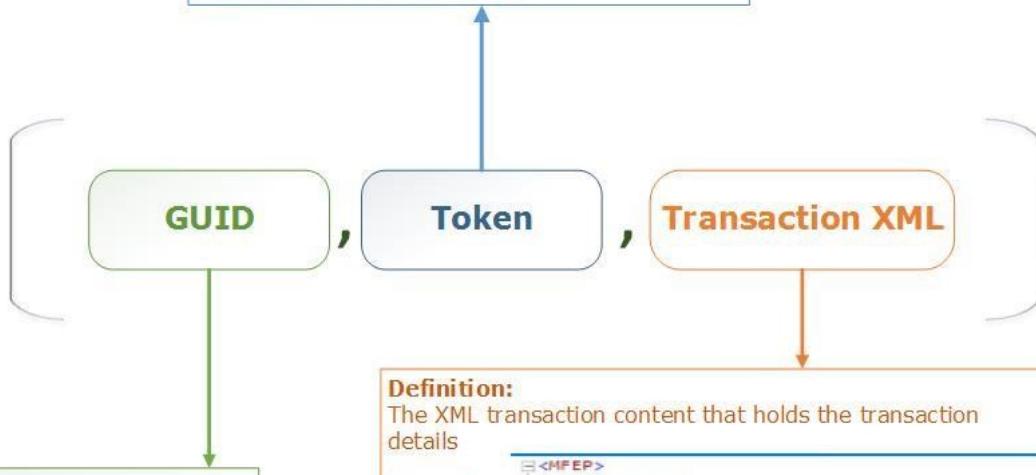


Con 2 Us

Integration Specs

Definition:
 The Token that was given to the participant as a result of the authentication process

Example:
 B44827BEDC0655A5D28FBFBE131216087AD5D5
 B824D32F0AFB9155BBD839D2E84



Definition:
 A (Global Unique Identifier) key that is generated from participant side or eFAWATEERcom side and is sent in each single request to identify the transaction

Example:
 b25992b5-7bac-44e4-906c-936172990940

Definition:
 The XML transaction content that holds the transaction details

Example:

```

<MFEP>
  <MsgHeader>
    <TmStp>
      <!--DD/MM/YYYY-SS:MM:HH-->
    </TmStp>
    <TrsInf>
      <SdrCode>
        <!--002-->
      </SdrCode>
      <RcvCode>
        <!--001-->
      </RcvCode>
      <ReqTyp>
        <!--BILUPRQ-->
      </ReqTyp>
    </TrsInf>
    <Sequence>
      <Sess>
        <!--Session Number-->
      </Sess>
      <Seq>
        <!--Sequence Number-->
      </Seq>
    </Sequence>
  </MsgHeader>
  <MsgBody>...</MsgBody>
  <MsgFooter>
    <Extra>
      <Customfield>
        <FieldID></FieldID>
        <FieldName></FieldName>
        <value></value>
        <ParentTagXPath></ParentTagXPath>
      </Customfield>
    </Extra>
    <Security>
      <Signature>
        <!--Signature Value-->
      </Signature>
      <SigXpath>
        <!--XPath for the signed element-->
      </SigXpath>
      <ChkSum>
        <!--Check Sum Value-->
      </ChkSum>
    </Security>
  </MsgFooter>
</MFEP>
  
```

4.18 Participants Authentication with ONEIC-TASDEED

- 4.18.1 For integration with ONEIC-TASDEED switch and to perform any operation over the ONEIC-TASDEED solution, the participant should be authorized by performing a regular handshaking operation as the following:
- Request for authentication
 - Receive the authentication Token
 - Use it for any future operation
- 4.18.2 The following diagram shows the steps for Authenticating with ONEIC-TASDEED solution:

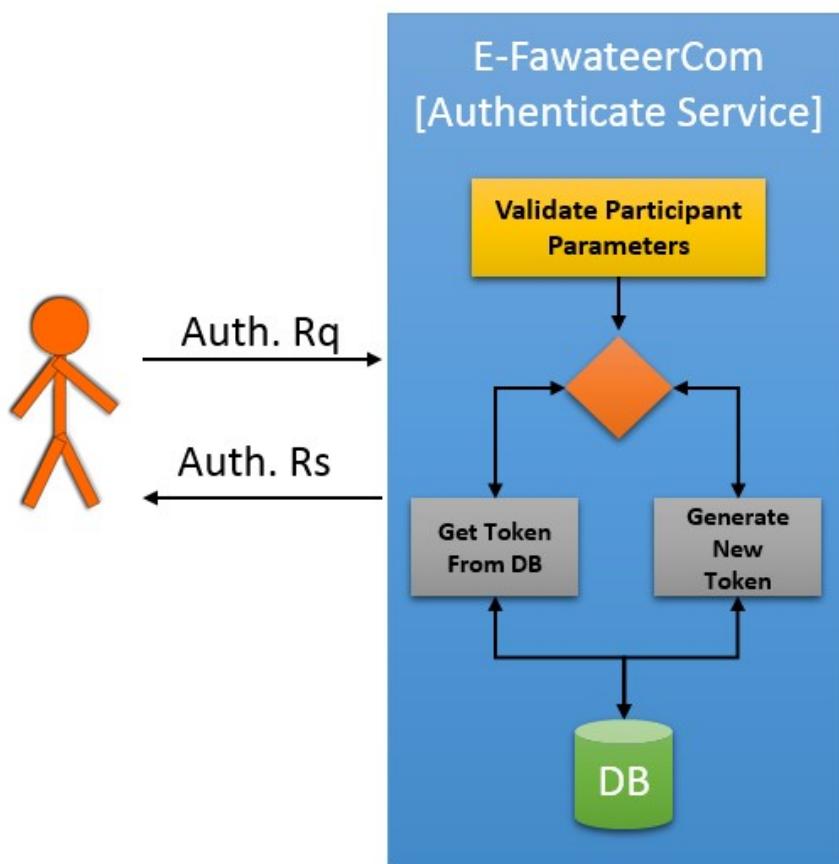


Figure 22 ONEIC-TASDEED Autehnentication Process



C o n 2 U s
Integration Specs

- 4.18.3 After the authentication operation is performed for a participant, any operation from that participant side will be successfully executed, if the authentication is not done, the service request will be rejected.



Con 2 Us
Integration Specs

Authentication (Token) Details

Operation Name: Authenticate

Called by: Banks/PSPs/Billers

Request Parameters:

Parameter Name	Description	Optional	Type	Sample
CustomerCode	The code given to the participant once generated	No	Integer	421
Password	The password given to the participant once registered in ONEIC-TASDEED	No	String	Newpassword 2349
GUID	Represents a Global Unique Identifier key generated by the Participant or ONEIC-TASDEED and used to identify the transaction	No	GUID	8328c40d- 7064-4d90- 9b2c4254a43dbcacf

Response Parameters:

Element Name	Optional
TokenConf	No
TokenKey	No





C o n 2 U s
Integration Specs

ExpiryDate	No
End of 'TokenConf'	

Request Sample:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:r="http://schemas.xmlsoap.org/ws/2005/02/rm"
             xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <r:Sequence s:mustUnderstand="1">
      <r:Identifier>urn:uuid:25c00033-ea23-4b32-946f-eeac0f2a8f0c</r:Identifier>
      <r:MessageNumber>1</r:MessageNumber>
    </r:Sequence>
    <a:Action s:mustUnderstand="1">http://tempuri.org/ITokenService/Authenticate</a:Action>
    <a:MessageID>urn:uuid:ddc89f2e-7d2e-4ccc-9f61-844e7a8999a1</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">http://172.20.0.22:7011/Token.svc</a:To>
  </s:Header>
  <s:Body>
    <Authenticate xmlns="http://tempuri.org/">
      <guid>924e7261-c072-4a1b-8f2d-b0f28b2bb235</guid>
      <customerCode>27</customerCode>
      <password>18DCA598D20093680EF78000DE35E0B87</password>
    </Authenticate>
  </s:Body>
</s:Envelope>
```

Figure 23 Authenticate Request Sample

Response Sample:





Con 2 Us

Integration Specs

```
<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T09:52:20</TmStp>
    <GUID>zx974b4e-4d7f-47b1-b8b7-2effc2ac911b</GUID>
    <TrsInf>
      <RcvCode>2</RcvCode>
      <ResTyp>TOKENRS</ResTyp>
    </TrsInf>
    <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </MsgHeader>
  <MsgBody>
    <TokenConf>
      <TokenKey>abchkdkabl+-0/aLP2HMXzrqBthRdR7tSnRftbDw9ibbGfr0lxAyR1Zskse8Sg0322</TokenKey>
      <ExpiryDate>2014-12-25T09:52:20</ExpiryDate>
    </TokenConf>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>abchkdkabl+-Oasdfasdfasdfsadfasdfasdfsawerqwer</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 24 Authenticate Response Sample

4.19 Digital Signature

- 4.19.1 A digital signature authenticates the XML messages in a similar manner a handwritten signature authenticates printed documents. This signature cannot be forged and it asserts that a named participant (Biller, Bank, PSP, or ONEIC-TASDEED) written or otherwise agreed to the message to which the signature is attached.
- 4.19.2 The recipient of a digitally signed message can verify that the message originated from the participant whose signature is attached to the message and that the message has not been altered either intentionally or accidentally since it was signed.
- 4.19.3 In addition, the signer of a XML message cannot later disown it by claiming that the signature was forged. In other words, digital signatures enable the 'Authentication' and 'Non-Repudiation' of digital messages, assuring the recipient



C o n 2 U s
Integration Specs

of a digital message of both the identity of the sender and the integrity of the message.

- 4.19.4 The participants and ONEIC-TASDEED should digitally sign and verify the signature value in all requests and responses for each process based on a pre-defined XPath <MsgBody> and not the full XML file.
- 4.19.5 ONEIC-TASDEED will provide Banks, PSPs and Billers with its digital certificate that contains only the public key which is based on PKI infrastructure that supports a key length of 2048 bit and signature hash algorithm SHA-2.
- 4.19.6 Further details about the certificate requirements will be shared in a separate document.

The signing and verifying process provided is as follows:

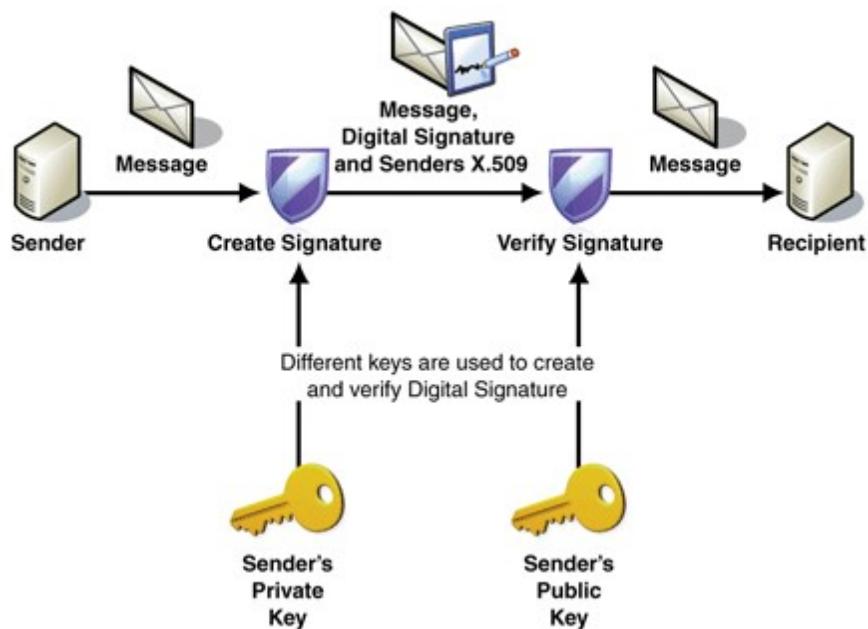


Figure 25 Signing & Verifying Process

4.20 Header and Footer

- 4.20.1 ONEIC-TASDEED Switch is designed to have almost a common request/response schemas for most of its services for the easiness of parsing and manipulating the XML data regardless building the request or parsing the response.
- 4.20.2 All services in the solution excluding prepaid services share a common Header details.
- 4.20.3 Prepaid Services separately share a common Header details.
- 4.20.4 The difference between the Header details of prepaid services and all other services is only in GUID; prepaid services contains 'GUID' element in Header of its Request messages while other services do not.
- 4.20.5 All services in the solution including prepaid services share a common Footer details.
- 4.20.6 For all messages, the Header, Body, and Footer (MsgHeader, MsgBody, MsgFooter) must be contained within 'MFEP' container.

```
<MFEP>
  <MsgHeader></MsgHeader>
  <MsgBody></MsgBody>
  <MsgFooter></MsgFooter>
</MFEP>
```

Figure 26 General Message Structure Schema

- 4.20.7 The 'Message Body' changes based on the operation that is being processed.

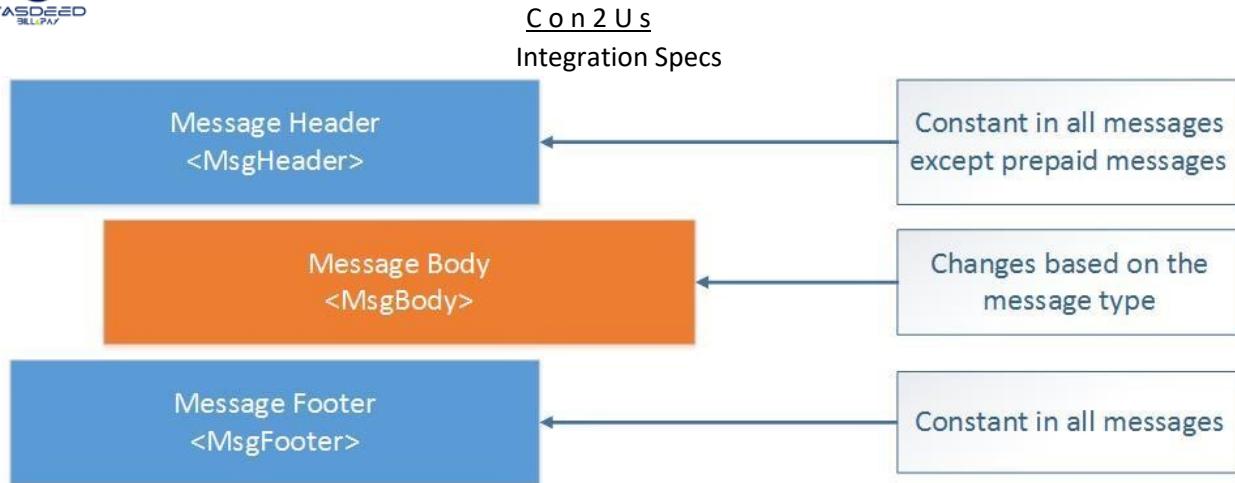


Figure 27 Message Structure

4.20.8 Header Details Request

Parameters:

Element Name	Mandatory/Conditional/Optional
TmStp	M
GUID	C
TrsInf	M
SdrCode	C
RcvCode	M
ReqTyp	M
End of 'TrsInf'	
Sequence	O



C o n 2 U s
Integration Specs

Sess	M
Seq	M
End of 'Sequence'	

Response Parameters:

Element Name	Mandatory/Conditional/Optional
TmStp	M
GUID	M
TrsInf	M
SdrCode	C
RcvCode	C
ResTyp	M
End of 'TrsInf'	
Sequence	O
Sess	M
Seq	M
End of 'Sequence'	
Result	M
ErrorCode	M
ErrorDesc	M



C o n 2 U s
Integration Specs

Severity	M
End of 'Result'	

Request Sample:

```
<MsgHeader>
<TmStp>2013-12-25T12:53:00</TmStp>
<TrsInf>
  <SdrCode>2</SdrCode>
  <RcvCode>1</RcvCode>
  <ReqTyp>BILPMTRQ</ReqTyp>
</TrsInf>
</MsgHeader>
```

Figure 28 Header Request Sample

Response Sample:

```
<MsgHeader>
<TmStp>2013-12-25T12:53:20</TmStp>
<GUID>c2967db5-25c5-4d71-9a85-f82f0649ded3</GUID>
<TrsInf>
  <RcvCode>2</RcvCode>
  <ResTyp>BILPMTRS</ResTyp>
</TrsInf>
<Result>
  <ErrorCode>0</ErrorCode>
  <ErrorDesc>Success</ErrorDesc>
  <Severity>Info</Severity>
</Result>
</MsgHeader>
```

Figure 29 Header Response Sample





Con 2 Us
Integration Specs

4.20.9 Footer Details Request

Parameters:

Element Name	Mandatory/Conditional/Optional
Extra	O
CustomField- Repeatable	M
FieldID	M
FieldName	M
Value	M
ParentTagXPath	M
End of 'CustomField'	
End of 'Extra'	
Security	M
Signature	C
End of 'Security'	

Response Parameters:

Element Name	Mandatory/Conditional/Optional
Extra	O
CustomField- Repeatable	M
FieldID	M





C o n 2 U s
Integration Specs

FieldName	M
Value	M
ParentTagXPath	M
End of 'CustomField'	
End of 'Extra'	
Security	M
Signature	C
End of 'Security'	





Request Sample:

```
<MsgFooter>
  <Security>
    <Signature>abchkdkabl++0/aLP2HMXzrqBthRdR7tSnRFtbDw9ipNGfr0lxAyRlZskse8Sg0as</Signature>
  </Security>
</MsgFooter>
```

Figure 30 Footer Request Sample

Response Sample:

```
<MsgFooter>
  <Security>
    <Signature>P2HMKdkabl++0/aLP2HMXzrqBthRdR7tabchkdkabl++0/aLP2HMXzrqBthRSnRFtbDw9ipNGfr0lxAyRlZskse8Sg0as</Signature>
  </Security>
</MsgFooter>
```

Figure 31 Footer Response Sample

4.21 Result Definition

- 4.21.1 When ONEIC-TASDEED or any participant generates any response message, this message will be filled into a custom xml message, this message will contain 'Result' container in 'MsgHeader' and/or 'MsgBody'; the Result in MsgHeader represents the status of the whole message, while the Result in MsgBody represents the status of record.
- 4.21.2 'Result' container contains a sub elements to display the error code, error description, and severity of the error.
- 4.21.3 The receiver of this message must check first the 'Result' status in 'MsgHeader' then check 'Result' status in 'MsgBody' to determine the appropriate way for handling this response.





C o n 2 U s

Integration Specs

The following figures displays response samples with 'Result' container:

```
<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T12:53:10</TmStp>
    <GUID>c2967db5-25c5-4d71-9a85-f82f0649ded3</GUID>
    <TrsInf>
      <RcvCode>2</RcvCode>
      <ResTyp>ACCTINQRS</ResTyp>
    </TrsInf>
    <Result>
      <ErrorCode>002</ErrorCode>
      <ErrorDesc>Invalid Signature</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </MsgHeader>
</MFEP>
```

Figure 32 Response with Result in Header Sample





Con 2 Us

Integration Specs

```
<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T09:52:20</TmStp>
    <GUID>zx974b4e-4d7f-47b1-b8b7-2effc2ac911b</GUID>
    <TrsInf>
      <RcvCode>2</RcvCode>
      <ResTyp>ACCTUPRS</ResTyp>
    </TrsInf>
    <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </MsgHeader>
  <MsgBody>
    <Accts>
      <Acct>
        <IdType>PAS</IdType>
        <Id>21341234345</Id>
        <Nation>UA</Nation>
        <Result>
          <ErrorCode>0</ErrorCode>
          <ErrorDesc>Success</ErrorDesc>
          <Severity>Info</Severity>
        </Result>
        <JOEPPSNo>21</JOEPPSNo>
      </Acct>
    </Accts>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>abchkdkabl+-0/aLP2HMXzrqBthRdR7tSnRftbDw9ibbGfr0lxAyRlZskse8Sg0322</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 33 Response with Result in Body Sample

4.22 Enumerated Types

ONEIC-TASDEED solution uses several enumerated types (Opened and Closed) to make sure that the system is consistent and dynamic at the same time by forcing communication protocols to have certain values, where these values are used to determine the solution behavior, do certain calculations and validations, and bring up related results accordingly.





Con 2 Us

Integration Specs

Enumerated types are used as parameters in the ONEIC-TASDEED web services and processes and are validated by ONEIC-TASDEED switch.

[Opened Enumerator]

Can be modified

[Closed Enumerator]

Cannot be modified

4.22.1 Operations Types

Operations that can be performed as part of the STP processes over ONEIC-TASDEED switch, either it is a Request 'RQ' or Response 'RS', and either the operation is initiated by ONEIC-TASDEED or participants, <ReqTyp> and <ResTyp> elements.

[Opened Enumerator]

Code	Description	Sender	Receiver	Online	Offline
ACCTINQRQ	Account Inquiry Request	Bank/PSP	ONEIC-TASDEED	T	
ACCTINQRS	Account Inquiry Response	ONEIC-TASDEED	Bank/PSP	T	
ACCTUPRQ	Account Upload Request	Bank/PSP	ONEIC-TASDEED	T	
ACCTUPRS	Account Upload Response	ONEIC-TASDEED	Bank/PSP	T	
ADCUSBILRQ	Add Customer Billing Request	Bank/PSP	ONEIC-TASDEED	T	
ADCUSBILRS	Add Customer Billing Response	ONEIC-TASDEED	Bank/PSP	T	

BILINQRQ	Bill Inquiry Request	Bank/PSP	ONEIC-TASDEED	T	
BILINQRS	Bill Inquiry Response	ONEIC-TASDEED	Bank/PSP	T	





Con 2 Us
Integration Specs

BILPMTRO	Bill Payment Request	Bank/PSP	ONEIC-TASDEED	T	
BILPMTRS	Bill Payment Response	ONEIC-TASDEED	Bank/PSP	T	
BILPULRQ	Bill Pull Request	ONEIC-TASDEED	Biller	T	
BILPULRS	Bill Pull Response	Biller	ONEIC-TASDEED	T	
BILRPREPADVALRQ	Biller Prepaid Validation Request	ONEIC-TASDEED	Biller	T	
BILRPREPADVALRS	Biller Prepaid Validation Response	Biller	ONEIC-TASDEED	T	
BILRSLSTRS	Billers List Response	ONEIC-TASDEED	Bank/PSP	T	
BILUPBATRQ	Bill Upload Batch Request	Biller	ONEIC-TASDEED		T
BILUPBATRS	Bill Upload Batch Response	ONEIC-TASDEED	Biller		T
BILUPRQ	Bill Upload Request	Biller	ONEIC-TASDEED	T	
BILUPRS	Bill Upload Response	ONEIC-TASDEED	Biller	T	
BLRPMTNTFRQ	Biller Payment Notification Request	ONEIC-TASDEED	Biller	T	T
BLRPMTNTFRS	Biller Payment Notification Response	Biller	ONEIC-TASDEED	T	T
BNKPMTNTFRQ	Bank Payment Notification Request	ONEIC-TASDEED	'Settlement Bank'	T	T
BNKPMTNTFRS	Bank Payment Notification Response	'Settlement Bank'	ONEIC-TASDEED	T	T





Con 2 Us
Integration Specs

PMTACKRQ	Payment Acknowledgement Request	ONEIC-TASDEED	'PSP Settlement Bank' / 'PSP' / 'Paying Bank'	T	T
PMTACKRS	Payment Acknowledgement Response	'PSP Settlement Bank' / 'PSP' / 'Paying Bank'	ONEIC-TASDEED	T	T
PMTINQRQ	Payment Inquiry Request	Bank/PSP	ONEIC-TASDEED	T	
PMTINQRS	Payment Inquiry Response	ONEIC-TASDEED	Bank/PSP	T	
PREPADPMTRQ	Prepaid Payment Request	Bank/PSP	ONEIC-TASDEED	T	
PREPADPMTRS	Prepaid Payment Response	ONEIC-TASDEED	Bank/PSP	T	
PREPADVALRQ	Prepaid Validation Request	Bank/PSP	ONEIC-TASDEED	T	
PREPADVALRS	Prepaid Validation Response	ONEIC-TASDEED	Bank/PSP	T	
RMVCUSBILRQ	Remove Customer Billing Request	Bank/PSP	ONEIC-TASDEED	T	
RMVCUSBILRS	Remove Customer Billing Response	ONEIC-TASDEED	Bank/PSP	T	
TOKNRS	Token Response	ONEIC-TASDEED	Bank/ PSP/ Biller	T	
SYSSVCAVARQ	System Service Availability Request	ONEIC-TASDEED	Bank/ PSP/ Biller	T	
BLRSVCAVARQ	Biller Service Availability Request	ONEIC-TASDEED	Bank/ PSP	T	

4.22.2 Billers Categories





C o n 2 U s
Integration Specs

The category under which billers can be classified, such as: Telecommunications, Electricity, etc... 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 ONEIC-TASDEED to associate messages with bills according to the associated service type.

The actual biller category is determined by participating Billers, but must be registered and approved by ONEIC-TASDEED before it is accepted, and it is managed through ONEIC-TASDEED Portal, <Category> or <BillerCategory> element.

[Opened Enumerator]

Code	Description
GOVT	Government Service
TELC	Telecommunication
INSR	Insurance
BKSV	Bank Service
MEDI	Medical
EDUC	Education
UTIL	Utility
ENTR	Entertainment
TRAN	Transportation
MDIA	Media
RLES	Real Estate Services
ASSO	Associations and Organizations and Foundations
ARLN	Air Line
CHRT	Charites



OTHR	Others
-------------	--------

4.22.3 Billers Integration Types

The integration type of how the biller is connected to ONEIC-TASDEED (Online, Offline, etc...) It should be approved by ONEIC-TASDEED, and it is managed through ONEIC-TASDEED Portal, <IntegrationType> element.

[Opened Enumerator]

Code	Description
Online	Biller is integrated online with the system
Offline	Biller is integrated offline with the system

4.22.4 Service Types

A service type assigns behavior to a bill, thus, permitting ONEIC-TASDEED to apply rules to bills in various manners. For example, a normal recurring bill requires different behavior at time of presentment and payment than an endorsement bill.

Billers must assign values of their choosing to each of their bills, but must first register these strings with ONEIC-TASDEED to avoid rejection, <ServiceType>element.

[Opened Enumerator]

Code	Description
FixedLine	A fixed Line
Internet	Internet
BroadBand	Broad Band Connection
Mobile	Mobile



Con 2 Us
Integration Specs

Electricity	Electricity
--------------------	-------------

4.22.5 Prepaid Categories

This enumerator describes the possible categories of prepaid services, <PrepaidCat> element.
[Opened Enumerator]

Code	Description
JD_1	Prepaid Category of JD_1
JD_1.5	Prepaid Category of JD_1.5
Mobile_3	Prepaid Category of Mobile_3
Cat_3	Prepaid Category of Cat_3
Cat_15	Prepaid Category of Cat_15

4.22.6 Bill Types

This enumerator defines the possible types of the bill being paid in terms of recurrence, <BillType> element.

[Closed Enumerator]

Code	Description
Recurring	Represents a normal recurring bill
OneOff	Represents a one-off bill

4.22.7 Bill Statuses





C o n 2 U s
Integration Specs

The Bill Status identifies the bill's statuses within ONEIC-TASDEED and is intended to be used when communicating bill data between separated systems, <BillStatus> element.

[Closed Enumerator]

Code	Description
BillNew	New Bill record
BillUpdated	Updated version of a bill
BillPaid	The bill is fully paid and the due amount is set to zero
BillPartiallyPaid	One or more payments have been applied to the bill, but the due amount still larger than zero
BillOverPaid	One or more payments have been applied to the bill, and the due amount became less than zero

4.22.8 Payment Types

This enumerator describes the payment types whether it is Postpaid or Prepaid, <PaymentType> element.

[Closed Enumerator]

Code	Description
Postpaid	Postpaid Payment Type
Prepaid	Prepaid Payment Type

4.22.9 Payment Statuses

This enumerator describes the possible payment statuses, <PmtStatus> element.





C o n 2 U s
Integration Specs

[Closed Enumerator]

Code	Description
PmtNew	New Payment record is inserted to the Payment database
PmtSent	Payment sent to CENTRAL BANK for settlement
PmtComplIt	Payment settlement is completed

4.22.10 Payment Sources

This enumerator describes the possible payment sources that used to determine the source of the payment record, <PmtSrc> element.

[Opened Enumerator]

Code	Description
BNKPmt	Bank Payment Transaction
PSPPmt	PSP Payment Transaction

4.22.11 Payment Record Types

This enumerator describes the possible values for payment records in reconciliation files that used to determine the type of the payment record, <PaymentsRecordType> element.

[Opened Enumerator]

Code	Description
PayingBank	Payment record data as 'Paying Bank'
SettlementBank	Payment record data as 'Settlement Bank'
PSPBank	Payment record data as 'PSP Settlement Bank'





C o n 2 U s
Integration Specs

PSP	Payment record data as 'PSP'
-----	------------------------------

4.22.12 Access Channels

The access channel is a channel where customers who owe bills can pay through, most of these channels are provided by Banks/PSPs, <AccessChannel> element.

[Opened Enumerator]

Code	Description
ATM	Automated Teller Machine
IVR	Interactive Voice Response
KIOSK	Kiosk Machines
INTERNET	Internet Browser
PORTAL	Portal
PSP	Payment Service Provider
BTELLER	Bank Teller
POS	Point Of Sale
CAM	Cash Acceptance Machine
CORP	Corporate
CCC	Call Center
PDA	Personal Digital Assistant
SMS	Short Message Service
JOMOPAY	Jordan Mobile Payment System





C o n 2 U s
Integration Specs

Mobile	Mobile
DDS	Direct Debit System

4.22.13 Payment Methods

This enumerator describes the possible payment methods used for paying bills, which means that a bill can be paid as cash, by credit card, etc... through the different access channels, <PaymentMethod> element.

[Opened Enumerator]

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

4.22.14 Payment Currencies

This enumerator describes the possible currencies used in ONEIC-TASDEED, <Currency> element.

[Opened Enumerator]

Code	Description
JOD	Jordan Dinar
USD	United States Dollar
EUR	Euro





C o n 2 U s
Integration Specs

4.22.15 Customers Official ID Types

This enumerator describes the possible values for the Customer ID Types, <IdType> element.
[Closed Enumerator]

Code	Description
NAT	National ID
PAS	Passport ID
IQA	Iqama ID
BIS	Business ID

4.22.16 Customers Nationalities

This enumerator describes the possible nationalities of customers to be defined in their profiles in ONEIC-TASDEED, <Nation> element.

[Closed Enumerator]

Code	Description
JO	Jordanian
IQ	Iraq
GB	British
KW	Kuwait

For the full list, you can refer to the section: Appendix 'Nationalities Codes and Descriptions'.

4.22.17 Customer Profile Statuses

This enumerator describes the possible statuses of a specific customer profile, and it can be managed through ONEIC-TASDEED Portal.





Con 2 Us
Integration Specs

[Closed Enumerator]

Code	Description
Active	Customer profile is Active
Inactive	Customer profile is Inactive

4.22.18 Billing Account Statuses

This enumerator describes the possible statuses of a specific billing account, and it can be managed through ONEIC-TASDEED Portal, <BillingStatus> element.

[Closed Enumerator]

Code	Description
Active	Billing Account is Active
Inactive	Billing Account is Inactive

4.22.19 Error Codes and Error Descriptions

ONEIC-TASDEED switch is responsible for validating all parameters in all messages, as the switch executes a set of validation processes to ensure that all information is correct and valid from security, syntax, semantic, etc... perspectives, and to achieve this, ONEIC-TASDEED used a set of Error Codes and corresponding Error Descriptions to communicate the result of validations with participants in order to be handled by them, <ErrorCode> and <ErrorDesc> elements.

[Opened Enumerators]

Error Code	Error Description
Info	0





Con 2 Us
Integration Specs

0	Success
Error	(1 - 999)
1	Invalid Token
2	Invalid Signature
3	Invalid Code Or Password
4	Expired Token
5	Invalid Sender Code
6	Invalid GUID
7	Duplicated GUID
101	Invalid XML Schema
303	Internal Error
304	Invalid Timestamp
306	Biller Not Found
307	Bank Not Found
308	Inactive Biller
309	Inactive Bank
310	Customer ID Or JOEBPPSNo Does Not Exist
311	Inactive Customer ID
312	CustomerID And JOEBPPSNo Should Not Be Sent Together
313	Inactive Billing Account
314	Billing Number Does Not Match Bill Number
315	Billing Does Not Exist Under This Profile
316	This Billing Already Exist Under This Profile





Con 2 Us
Integration Specs

317	Invalid Issue Date
318	Invalid Open Date
319	Invalid Due Date
320	Invalid Close Date
321	Invalid Expiry Date
322	Unrecognized Service Type
323	Bill Has Been Expired Or Closed
324	Bill Has Been Paid Previously
325	Invalid Paid Amount
326	Invalid Due Amount
327	Invalid Process Date
328	This Payment Does Not Belong To Any Bill
329	Bill Amount Is Not In A Correct Range
330	Invalid Lower Or Upper Value In Payment Constant
332	Maximum Number Of Transactions Reached
333	Maximum Amount Of Total Transactions Reached
334	Invalid Bill Type
335	Biller Type Is Pull thus Cannot Upload Bills
336	Total Of Sub Payments Does Not Equal Due Amount
337	Not Allowed To Send Sub Payments With Allow Partial
338	The CustId Or JOEBPPSNo Is Missing
339	Bill Open Date Or Due Date Is Not Reached Yet





C o n 2 U s
Integration Specs

340	Duplicated Payment
341	Duplicated Payments For Same Bill
342	Bill Has A Payment Under Processing
343	Duplicated Bill
344	Start Date Should Not Be Older Than Two Years From Today
345	End Date Should Not Be Larger Than Today
346	Failed To Read Bill Batch File

347	Invalid File Name Schema
350	This Settlement Bank Is Not Related To The Biller
351	Inactive Settlement Bank
352	Process Date Should Be Equal Or less Than TimeStamp
353	Paid Amount Exceeds The Upper Limit Of The Bill
355	Invalid Rec Count
356	Billing Number In Response Does Not Match Billing Number In Request
357	Bill Number Does Not Equal Billing Number
358	Invalid JOEBPPSTrx
359	Invalid STMT Date
360	Unsuccessful Payment Notification
362	Prepaid Service Type Is Not Allowed
363	Postpaid Service Type Is Not Allowed
364	Billing Number Is Mandatory For This Service
365	Validation Code In Response Does Not Match Validation Code In Request





C o n 2 U s
Integration Specs

366	Prepaid Cat In Response Does Not Match Prepaid Cat In Request
367	Unsuccessful Biller Prepaid Validation
368	Not Allowed To Send Prepaid Cat
370	This Payment Does Not Belong To Any Prepaid Validation Trx
371	Invalid Payment Type
372	Due Amt Should Not Be Sent If Contains Prepaid Cat
373	Prepaid Cat Is Missing
374	Due Amt Should Be Sent If Does Not Contain Prepaid Cat
375	This Service Type Should Not Have Prepaid Cat
376	Not Allow To Send Billing Number For This Service Type
377	Unrecognized Prepaid Cat
379	Invalid Biller Code
380	Signature Element Is Missing

381	Inactive PSP
382	PSP Not Found
383	This Inquiry Does Not Belong To Any Payment
384	Service Elements Sent In Request Does Not Match Service Specifications
385	GUID Sent In Request Does Not Equal GUID Sent As Parameter
386	Expired Validation Code
387	Invalid Access Channel
388	Start Date Should be Less Than Or Equal End Date
394	Advanced Payment Is Not Allowed If AllowPart Is False





C o n 2 U s
Integration Specs

395	In Valid Advanced Paid Amount
396	Advanced Payment Is Not Allowed In Sub Payments
397	Billing Number Reached Its Max Amount
398	Billing Number Reached Its Max Transaction
399	Billing Number Will Exceed Its Max Amount
400	Billing Number Will Exceed Its Max Transaction
401	Bill Not Found
402	Customer ID And JOEBPPSNo Should Not Be Null When You Send Payer Info
403	Customer ID Missing Some Required items
404	Error Sent By Biller
405	Not Allowed To Send More Than One Sub Payment With Allow Partial
701	Reversed Payment
702	InqRefNo Is Required
703	Not Allowed To Send InqRefNo
704	InqRefNo Does Not Exist
705	This InqRefNo Already Got A Payment



706	Payer Info Is Required
707	Not Allowed To Send This Payment For Non Jordanian Customers
708	Invalid Customer ID
750	This Bill Should Have Sub Payments
751	Invalid Settlement Bank Code In Sub Payments
752	Should Not Sent Different Settlement Bank Codes In Sub Payments
753	Should Not Sent Different BIC Codes In Sub Payments
754	Account BIC Code Does Not Match Settlement Bank BIC Code
755	In Active Settlement Bank In SubPayments

4.22.20 Severity Types

This enumerator is used in 'Result' container for Response messages to describe the severity of error, <Severity> element.

[Closed Enumerator]

Code	Description
Error	The message was not processed successfully. The error code and description indicate the cause of the error
Info	This status indicates the server accepted and processed the message successfully. The code and description provide additional information

4.22.21 Biller Service Availability Status

This enumerator is used in <Status> element that exists into Biller Service Availability Request to describe the status of the Biller, Or Biller Services at that moment.

[Closed Enumerator]



C o n 2 U s
Integration Specs

Code	Description
Up	The Biller/ Service is Up and Running.
Down	The Biller/ Service is Down or Unreachable.

4.22.22 System Service Availability Status

This enumerator is used in <Status> element that exists into System Service Availability Request to describe the status of the ONEIC-TASDEED system.

[Opened Enumerators]

Code	Description
Up	ONEIC-TASDEED system is available.
Down	ONEIC-TASDEED system is unavailable due to maintenance or upgrade.

5. ONEIC-TASDEED Integration Services

ONEIC-TASDEED switch service is a collection of related Request and Response aggregates. Each service request initiates a series of actions on a server whereas the response provides feedback to a corresponding request. ONEIC-TASDEED currently supports only one request/response message for each service. This may change in the future for efficiency improvement.

ONEIC-TASDEED services are divided by the beneficiary of the service, either Banks/PSPs, or Billers.

Notes:

- Samples used in the operations below do not include probably the optional parameters for the easiness of elaboration.



C o n 2 U s

Integration Specs

- In message structure tables below, all rows highlighted in 'blue' represents elements from 'Container' data type.

5.1 Billers List Service

The Billers List service permits Banks/PSPs to get the latest active billers information in ONEIC-TASDEED. Bank/PSP may initiate this service at any time of their choosing by sending a request to ONEIC-TASDEED.

Operation Name: GetBillersList

Called by: Banks/PSPs

Request Parameters:

In this service, the Bank/PSP is required to send only 'GUID' and 'Token Key' as parameters with no XML message, and ONEIC-TASDEED will respond by XML response.

Parameter Name	Mandatory/Conditional/Optional
GUID	M
Token Key	M

Response Parameters:

Element Name	Mandatory/Conditional/Optional
RecCount	M
SystemMarketing	O
EnMsg	O
ArMsg	O

C o n t a i n e r s
Integration Specs

EnImg	O
ArImg	O
End of 'SystemMarketing' container	
BillersRec	M

BillerRec - Repeatable	M
BillerInfo	M
Code	M
IntegrationType	M
StmtBankCode	M
Website	M
Email	M
Phone	M
End of 'BillerInfo' container	
BillerName	M
EnShortName	M
ArShortName	M
EnName	M
ArName	M
End of 'BillerName' container	

C o n t a i n e r s
Integration Specs

BillerMarketing	O
EnLogo	O
ArLogo	O
EnMsg	O
ArMsg	O
EnImg	O
ArImg	O

End of 'BillerMarketing' container	
BillerCategory	M
EnShortName	M
ArShortName	M
EnName	M
ArName	M
End of 'BillerCategory' container	
BillerServices	M
BillerService	M
ServiceInfo	M
Code	M
Type	M

C o n t a i n e r s
Integration Specs

EnShortDesc	M
ArShortDesc	M
EnDesc	M
ArDesc	M
PaymentType	M
BillingNoRequired	C
ContainsPrepaidCats	C

End of 'ServiceInfo' container

ServiceMarketing	O
EnMsg	O

ArMsg	O
EnImg	O
ArImg	O

End of 'ServiceMarketing' container

BillingInfo	C
EnShortDesc	M
ArShortDesc	M
EnDesc	M
ArDesc	M

C o n t a i n e r s
Integration Specs

EnImg	O
ArImg	O
End of 'BillingInfo' container	
PrepaidCategories	C
PrepaidCatInfo - Repeatable	M
Code	M
Type	M
EnShortDesc	M
ArShortDesc	M
EnDesc	M
ArDesc	M
End of 'PrepaidCatInfo' container	
End of 'PrepaidCategories' container	
End of 'BillerService' container	
End of 'BillerServices' container	
End of 'BillerRec' Container	
End of 'BillersRec' Container	

C o n 2 U s

Integration Specs

Response Sample:

```

<MFEP xmlns="">
  - <MsgHeader>
    - <TmStp>2016-01-04T09:31:52</TmStp>
    <GUID>2d4a3634-9c30-4e66-81f5-8ba9de9d93e5</GUID>
  - <TrlInf>
    - <RcvCode>2</RcvCode>
    <ResTyp>BILRSLSSTRS</ResTyp>
  </TrlInf>
  - <Result>
    <ErrorCode>0</ErrorCode>
    <ErrorDesc>Success</ErrorDesc>
    <Severity>Info</Severity>
  </Result>
</MsgHeader>
- <MsgBody>
  <RecCount>1</RecCount>
  - <SystemMarketing>
    <EnImg>http://www.mfep.com/sysimgen.jpg</EnImg>
    <ArImg>http://www.mfep.com/sysimgar.jpg</ArImg>
  </SystemMarketing>
  - <BillersRec>
    - <BillerRec>
      - <BillerInfo>
        <Code>15</Code>
        <IntegrationType>Online</IntegrationType>
        <StmtBankCode>7</StmtBankCode>
        <Website>http://www.testbillier.com.jo</Website>
        <Email>Info@testbillier.com.jo</Email>
        <Phone>+9620655312342</Phone>
      </BillerInfo>
      - <BillerName>
        <EnShortName>test biller</EnShortName>
        <ArShortName>بiller (العن)</ArShortName>
        <EnName>test biller corporation</EnName>
        <ArName>بiller (العن)</ArName>
      </BillerName>
      - <BillerCategory>
        <EnShortName>GOVT</EnShortName>
        <ArShortName>بiller</ArShortName>
        <EnName>Government sector</EnName>
        <ArName>بiller</ArName>
      </BillerCategory>
      - <BillerServices>
        - <BillerService>
          - <ServiceInfo>
            <Code>11</Code>
            <Type>Internet</Type>
            <EnShortDesc>Internet</EnShortDesc>
            <ArShortDesc>الإنترنت</ArShortDesc>
            <EnDesc>Internet services</EnDesc>
            <ArDesc>الإنترنت</ArDesc>
            <PaymentType>Prepaid</PaymentType>
            <BillingNoRequired>true</BillingNoRequired>
            <ContainsPrepaidCats>true</ContainsPrepaidCats>
          </ServiceInfo>
          - <BillingInfo>
            <EnShortDesc>Subscription No</EnShortDesc>
            <ArShortDesc>الرقم</ArShortDesc>
            <EnDesc>Please enter your Internet Subscription No</EnDesc>
            <ArDesc>الرقم</ArDesc>
          </BillingInfo>
          - <PrepaidCategories>
            - <PrepaidCatInfo>
              <Code>4</Code>
              <Type>1</Type>
              <EnShortDesc>Card 5</EnShortDesc>
              <ArShortDesc>5 جنية</ArShortDesc>
              <EnDesc>Internet Prepaid card 5JD</EnDesc>
              <ArDesc>5 جنية</ArDesc>
            </PrepaidCatInfo>
          </PrepaidCategories>
        </BillerService>
      </BillerServices>
    </BillerRec>
  </BillersRec>
</MsgBody>
- <MsgFooter>
  - <Security>
    <Signature>WV4JupNkoqHnVmC9M1FmCzEGTxPvaIsBMxJaTq+afJ6Grk9HI6/JY/+Td12j/I7Q==</Signature>
  </Security>
</MsgFooter>
</MFEP>

```

Figure 34 Billers List Response Sample

5.2 Account Upload Service (Customer Profile)

The Account Upload (Upload Customer Profile) service permits the Bank/PSP to manage all accounts (Customer Profiles) and Billing Accounts in ONEIC-TASDEED system. To this end, the Account Upload request contains a series of elements to create new account (customer profile).

Bank/PSP may initiate this service at any time of their choosing by sending the request message to ONEIC-TASDEED. All new accounts are created in an 'Active' status.

Operation Name: AccountUpload

Called by: Banks/PSPs

Request Parameters:

Element Name	Mandatory/Conditional/Optional
Accts	M
Acct – repeatable	M
IdType	M
Id	M
Nation	M
Name	O
Phone	O
Address	O
Email	O
End of 'Acct' container	
End of 'Accts' container	

Response Parameters:

Element Name	Mandatory/Conditional/Optional
Accts	M
Acct- repeatable	M
IdType	M
Id	M
Nation	M
Result	M
ErrorCode	M
ErrorDesc	M
Severity	M
End of 'Result' container	
JOEBPPSNo	C
End of 'Acct' container	
End of 'Accts' container	

Request Sample:

C o n 2 U s

Integration Specs

```
- <MFEP>
  - <MsgHeader>
    <TmStp>2016-01-04T11:30:09</TmStp>
    - <TrsInf>
      <SdrCode>2</SdrCode>
      <RcvCode>1</RcvCode>
      <ReqTyp>ACCTUPRQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  - <MsgBody>
    - <Accts>
      - <Acct>
        <IdType>PAS</IdType>
        <Id>1234444456789</Id>
        <Nation>JO</Nation>
        <Name>Mohammad Refai</Name>
        <Phone>+962785846344</Phone>
        <Address>Amman</Address>
        <Email>mrefai@madfooat.com</Email>
      </Acct>
    </Accts>
  </MsgBody>
  - <MsgFooter>
    - <Security>
      <Signature>BxI1W94QbKk3LOLbHDs8/DaA8P5jCDf6qki6/1rWGNRrSsA==</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 35 Account Upload Request Sample

Response Sample:



C o n 2 U s

Integration Specs

```
- <MFEP xmlns="">
  - <MsgHeader>
    <TmStp>2016-01-04T11:31:02</TmStp>
    <GUID>42784411-8001-4e8c-9229-9c06719743f5</GUID>
  - <TrsInf>
    <RcvCode>2</RcvCode>
    <ResTyp>ACCTUPRS</ResTyp>
  </TrsInf>
  - <Result>
    <ErrorCode>0</ErrorCode>
    <ErrorDesc>Success</ErrorDesc>
    <Severity>Info</Severity>
  </Result>
</MsgHeader>
- <MsgBody>
  - <Accts>
    - <Acct>
      <IdType>PAS</IdType>
      <Id>12344444456789</Id>
      <Nation>JO</Nation>
    - <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
    <JOEBPPSNo>599</JOEBPPSNo>
  </Acct>
  </Accts>
</MsgBody>
- <MsgFooter>
  - <Security>
    <Signature>HGt1W94QbKk3LOLbHDs6/JkoP8P5jCDH78qkO986/1rWGNRrSSDl g==</Signature>
  </Security>
</MsgFooter>
</MFEP>
```

Figure 36 Account Upload Response Sample

5.3 Account Inquiry Service

The Account Inquiry service permits Banks/PSPs to inquire about a specific Account (Customer Profile) and view its information and the Billing Accounts listed under it in ONEIC-TASDEED system.

Banks/PSPs may initiate this service at any time of their choosing by sending the request message to ONEIC-TASDEED.

Operation Name: AccountInquiry

Called by: Banks/PSPs

Request Parameters:



C o n 2 U s
Integration Specs

Element Name	Mandatory/Conditional/Optional
Acct	M
CustId	C
IdType	M
Id	M
Nation	M
End of 'CustId' container	
JOEBPPSNo	C
End of 'Acct' container	

Response Parameters:

Element Name	Mandatory/Conditional/Optional
Acct	M
Result	M

ErrorCode	M
ErrorDesc	M
Severity	M
End of 'Result' container	
CustId	M
IdType	M

C o n 2 U s
 Integration Specs

Id	M
Nation	M
Name	O
Phone	O
Address	O
Email	O
End of 'CustId' container	
JOEBPPSNo	M
RecCount	C
BillingsRec	C
BillingRec - repeatable	M
BillerCode	M
BillingNo	M
ServiceType	M
BillingStatus	M
NickName	O
End of 'BillingRec' container	
End of 'BillingsRec' container	
End of 'Acct' container	

Request Sample:

```
<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T12:53:00</TmStp>
    <TrsInf>
      <SdrCode>2</SdrCode>
      <RcvCode>1</RcvCode>
      <ReqTyp>ACCTINQRQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  <MsgBody>
    <Acct>
      <CustId>
        <IdType>PAS</IdType>
        <Id>723123412</Id>
        <Nation>AX</Nation>
      </CustId>
    </Acct>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>chkdkabl++0/aLP2HMXzrqBthRdR7tSnRFtbDw9ipNGfr0lxAyRlZskse8Sg0as</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 37 Account Inquiry Request Sample

Response Sample:

```
<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T12:53:10</TmStp>
    <GUID>c2967db5-25c5-4d71-9a85-f82f0649ded3</GUID>
    <TrsInf>
      <RcvCode>2</RcvCode>
      <ResTyp>ACCTINQRS</ResTyp>
    </TrsInf>
    <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </MsgHeader>
  <MsgBody>
    <Acct>
      <Result>
        <ErrorCode>0</ErrorCode>
        <ErrorDesc>Success</ErrorDesc>
        <Severity>Info</Severity>
      </Result>
      <CustId>
        <IdType>PAS</IdType>
        <Id>21341234345</Id>
        <Nation>UA</Nation>
      </CustId>
      <JOEBPPSNo>21</JOEBPPSNo>
      <RecCount>1</RecCount>
      <BillingsRec>
        <BillingRec>
          <BillerCode>13</BillerCode>
          <BillingNo>079999999999</BillingNo>
          <ServiceType>FixedLine</ServiceType>
          <BillingStatus>Active</BillingStatus>
        </BillingRec>
      </BillingsRec>
    </Acct>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>12kdkabl++0/aLP2HMXzrqBthRdR7tSnRFtbDw9ipNGfr0lxAyRlZskse8Sg0a</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 38 Account Inquiry Response Sample

5.4 Add Customer Billing Service



C o n t a i n e r s

Integration Specs

The Add Customer Billing service permits the Bank/PSP to add Billing Accounts into specific customer profile in ONEIC-TASDEED system.

Banks/PSPs may initiate this service at any time of their choosing by sending the request message to ONEIC-TASDEED.

Operation Name: AddCustomerBilling

Called by: Banks/PSPs

Request Parameters:

Element Name	Mandatory/Conditional/Optional
Acct	M
JOEBPPSNo	M
BillingsRec	M
BillingRec	M
BillerCode	M
BillingNo	M
ServiceType	M
NickName	O
End of 'BillingRec' container	
End of 'BillingsRec' container	
End of 'Acct' container	

Response Parameters:



C o n 2 U s
Integration Specs

Element Name	Mandatory/Conditional/Optional
Acct	M
JOEBPPSNo	M
Result	M
ErrorCode	M
ErrorDesc	M
Severity	M
End of 'Result' container	
BillingsRec	C
BillingRec	M
BillerCode	M
BillingNo	M
ServiceType	M
NickName	O
Result	M
ErrorCode	M
ErrorDesc	M
Severity	M
End of 'Result' container	
End of 'BillingRec' container	



End of 'BillingsRec' container

End of 'Acct' container

Request Sample:

```
<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T12:53:20</TmStp>
    <TrsInf>
      <SdrCode>2</SdrCode>
      <RcvCode>1</RcvCode>
      <ReqTyp>ADCUSBILRQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  <MsgBody>
    <Acct>
      <JOEBPPSNo>3172312341232</JOEBPPSNo>
      <BillingsRec>
        <BillingRec>
          <BillerCode>13</BillerCode>
          <BillingNo>07999999999</BillingNo>
          <ServiceType>Electricity</ServiceType>
        </BillingRec>
      </BillingsRec>
    </Acct>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>126hkdkvel++0/aLP2HMXzrqBthRdR7tSnRFtbDw9ipNGfr0lxAyRlZskse+Ixm0</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 39 Add Customer Billing Request Sample

Response Sample:



C o n 2 U s

Integration Specs

```
<MFEP>
<MsgHeader>
<TmStp>2013-12-25T12:53:30</TmStp>
<GUID>12967db5-25c5-4d71-9a85-f82f0649ded3</GUID>
<TrsInf>
<RcvCode>2</RcvCode>
<ResTyp>ADCUSBILRS</ResTyp>
</TrsInf>
<Result>
<ErrorCode>0</ErrorCode>
<ErrorDesc>Success</ErrorDesc>
<Severity>Info</Severity>
</Result>
</MsgHeader>
<MsgBody>
<Acct>
<JOEBPPSNo>3172312341232</JOEBPPSNo>
<Result>
<ErrorCode>0</ErrorCode>
<ErrorDesc>Success</ErrorDesc>
<Severity>Info</Severity>
</Result>
<BillingsRec>
<BillingRec>
<BillerCode>13</BillerCode>
<BillingNo>07999999999</BillingNo>
<ServiceType>Electricity</ServiceType>
<Result>
<ErrorCode>0</ErrorCode>
<ErrorDesc>Success</ErrorDesc>
<Severity>Info</Severity>
</Result>
</BillingRec>
</BillingsRec>
</Acct>
</MsgBody>
<MsgFooter>
<Security>
<Signature>2kdkabl++0/aLP2HMXzrqBthRdR7tSnRtbDw9ipNGfr0lxAyRlZskse8Scfg</Signature>
</Security>
</MsgFooter>
</MFEP>
```

Figure 40 Add Customer Billing Response Sample

5.5 Remove Customer Billing Service

The Remove Customer Billing service permits the Bank/PSP to remove Billing Accounts from specific customer profile in ONEIC-TASDEED system.

Banks/PSPs may initiate this service at any time of their choosing by sending the request message to ONEIC-TASDEED.

C o n 2 U s
Integration Specs

Operation Name: RemoveCustomerBilling

Called by: Banks/PSPs

Request Parameters:

Element Name	Mandatory/Conditional/Optional
Acct	M
JOEBPPSNo	M
BillingsRec	M
BillingRec	M
BillerCode	M
BillingNo	M
ServiceType	M
End of 'BillingRec' container	
End of 'BillingsRec' container	
End of 'Acct' container	

Response Parameters:

Element Name	Mandatory/Conditional/Optional
Acct	M
JOEBPPSNo	M
Result	M
ErrorCode	M



C o n t a i n e r s
Integration Specs

ErrorDesc	M
Severity	M
End of 'Result' container	
BillingsRec	C
BillingRec	M
BillerCode	M
BillingNo	M
ServiceType	M
Result	M
ErrorCode	M
ErrorDesc	M
Severity	M
End of 'Result' container	
End of 'BillingRec' container	
End of 'BillingsRec' container	
End of 'Acct' container	

Request Sample:



C o n 2 U s

Integration Specs

```
<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T09:52:20</TmStp>
    <TrsInf>
      <SdrCode>2</SdrCode>
      <RcvCode>1</RcvCode>
      <ReqTyp>RMVCUSBILRQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  <MsgBody>
    <Acct>
      <JOEBPPSNo>3172312341232</JOEBPPSNo>
      <BillingsRec>
        <BillingRec>
          <BillerCode>13</BillerCode>
          <BillingNo>07999999999</BillingNo>
          <ServiceType>Electricity</ServiceType>
        </BillingRec>
      </BillingsRec>
    </Acct>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>abchkdkabl+-0/aLP2HMXzrqBthRdR7tSnRFtbDw9ibbGfr0lxAyR1Zskse8Sg0322</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 41 Remove Customer Billing Request Sample

Response Sample:

C o n 2 U s

Integration Specs

```

<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T09:52:23</TmStp>
    <GUID>zx974b4e-4d7f-47b1-b8b7-2effc2ac911b</GUID>
    <TrsInf>
      <RcvCode>2</RcvCode>
      <ResTyp>RMVCUSBILRS</ResTyp>
    </TrsInf>
    <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </MsgHeader>
  <MsgBody>
    <Acct>
      <JOEBPPSNo>3172312341232</JOEBPPSNo>
      <Result>
        <ErrorCode>0</ErrorCode>
        <ErrorDesc>Success</ErrorDesc>
        <Severity>Info</Severity>
      </Result>
    <BillingsRec>
      <BillingRec>
        <BillerCode>13</BillerCode>
        <BillingNo>079999999999</BillingNo>
        <ServiceType>Electricity</ServiceType>
        <Result>
          <ErrorCode>0</ErrorCode>
          <ErrorDesc>Success</ErrorDesc>
          <Severity>Info</Severity>
        </Result>
      </BillingRec>
    </BillingsRec>
  </Acct>
</MsgBody>
<MsgFooter>
  <Security>
    <Signature>a2c6hkdkvel++0/aLP2HMXzrqBthRdR7tSnRftbDw9ipNGfr0lxAyRlZskse+Ixm0</Signature>
  </Security>
</MsgFooter>
</MFEP>

```

Figure 42 Remove Customer Billing Response Sample

5.6 Bill Upload Service and Bill Batch Service

The Bill Upload and Bill Batch services permits the efficient transfer of bill data from Biller billing applications to ONEIC-TASDEED either online or offline, and allows the biller to send a bulk of bills in the same request. In addition, it allows the biller to upload the bill details



C o n 2 U s

Integration Specs

including the bill amount break down details -if any- (many sub-payments to different Banking accounts that make the total bill amount).

Bill Upload process divided into two-separated services:

1. Bill Upload Service
 - **Operation Name:** UploadBill
2. Bill Batch Service
 - **Through FTP/SFTP Server**

Called by: Billers

Request Parameters:

Element Name	Mandatory/Conditional/Optional
BillsRec	M
BillRec - repeatable	M
AcctInfo	M
BillingNo	M
BillNo	M
End of 'AcctInfo' container	
BillStatus	M
DueAmount	M
IssueDate	M
OpenDate	O
DueDate	M
ExpiryDate	O
CloseDate	O



C o n 2 U s
Integration Specs

ServiceType	M
BillType	O
PmtConst	O
AllowPart	M
Lower	M
Upper	M
End of 'PmtConst' container	
SubPmts	C
SubPmt – repeatable	M
Amount	M
SetBnkCode	M
AcctNo	M
End of 'SubPmt' container	
End of ' SubPmts' container	
End of 'BillRec' container	
End of 'BillsRec' container	

Response Parameters:

Element Name	Mandatory/Conditional/Optional
BillsRec	M

C o n 2 U s
Integration Specs

BillRec - repeatable	M
AcctInfo	M
BillingNo	M
BillNo	M
End of 'AcctInfo' container	
ServiceType	M
Result	M
ErrorCode	M
ErrorDesc	M
Severity	M
End of 'Result' container	
End of 'BillRec' container	
End of 'BillsRec' container	

Request Sample:

C o n 2 U s

Integration Specs

```
<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T08:52:59</TmStp>
    <TrsInf>
      <SdrCode>4</SdrCode>
      <RcvCode>1</RcvCode>
      <ReqTyp>BILUPRQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  <MsgBody>
    <BillsRec>
      <BillRec>
        <AcctInfo>
          <BillingNo>1325646</BillingNo>
          <BillNo>1325646</BillNo>
        </AcctInfo>
        <BillStatus>BillNew</BillStatus>
        <DueAmount>1000</DueAmount>
        <IssueDate>2013-12-25T08:52:59</IssueDate>
        <DueDate>2013-12-30T08:52:59</DueDate>
        <ServiceType>Custom</ServiceType>
      </BillRec>
    </BillsRec>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>cadhkdkabl+-0/aLP2HMXzrqBthRdR7tSnRFtbDw9ipNGfr0lxAyR1Zskse8Sg0456</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 43 Bill Upload Request Sample

Response Sample:



C o n 2 U s

Integration Specs

```
<MFEP>
<MsgHeader>
  <TmStp>2013-12-25T09:52:50</TmStp>
  <GUID>2x5179aB-1525-2168-94F4-F1EA99BE1DFF</GUID>
  <TrsInf>
    <RcvCode>4</RcvCode>
    <ResTyp>BILUPRS</ResTyp>
  </TrsInf>
  <Result>
    <ErrorCode>0</ErrorCode>
    <ErrorDesc>Success</ErrorDesc>
    <Severity>Info</Severity>
  </Result>
</MsgHeader>
<MsgBody>
  <BillsRec>
    <BillRec>
      <AcctInfo>
        <BillingNo>1325646</BillingNo>
        <BillNo>1325646</BillNo>
      </AcctInfo>
      <ServiceType>Custom</ServiceType>
      <Result>
        <ErrorCode>0</ErrorCode>
        <ErrorDesc>Success</ErrorDesc>
        <Severity>Info</Severity>
      </Result>
    </BillRec>
  </BillsRec>
</MsgBody>
<MsgFooter>
  <Security>
    <Signature>aadhkdkablsd0/aLP2HMXzrqBthRdR7tSnRftbDw9ipNGfr0lxAyRlZskse8Sg04ff</Signature>
  </Security>
</MsgFooter>
</MFEP>
```

Figure 44 Bill Upload Response Sample

Request Sample:



C o n 2 U s

Integration Specs

```
<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T08:52:59</TmStp>
    <TrsInf>
      <SdrCode>4</SdrCode>
      <RcvCode>1</RcvCode>
      <ReqTyp>BILUPBATRQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  <MsgBody>
    <BillsRec>
      <BillRec>
        <AcctInfo>
          <BillingNo>1325646</BillingNo>
          <BillNo>1325646</BillNo>
        </AcctInfo>
        <BillStatus>BillNew</BillStatus>
        <DueAmount>1000</DueAmount>
        <IssueDate>2013-12-25T08:52:59</IssueDate>
        <DueDate>2013-12-30T08:52:59</DueDate>
        <ServiceType>Custom</ServiceType>
      </BillRec>
    </BillsRec>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>cadhkdkabl+-0/aLP2HMXzrqBthRdR7tSnRFtbDw9ipNGfr0lxAyRlZskse8Sg0456</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 45 Bill Batch Request Sample

Response Sample:



C o n 2 U s

Integration Specs

```
<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T09:52:50</TmStp>
    <GUID>2x5179aB-1525-2168-94F4-F1EA99BE1DFF</GUID>
    <TrsInf>
      <RcvCode>4</RcvCode>
      <ResTyp>BILUPBATRS</ResTyp>
    </TrsInf>
    <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </MsgHeader>
  <MsgBody>
    <BillsRec>
      <BillRec>
        <AcctInfo>
          <BillingNo>1325646</BillingNo>
          <BillNo>1325646</BillNo>
        </AcctInfo>
        <ServiceType>Custom</ServiceType>
        <Result>
          <ErrorCode>0</ErrorCode>
          <ErrorDesc>Success</ErrorDesc>
          <Severity>Info</Severity>
        </Result>
      </BillRec>
    </BillsRec>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>aadhkdkablsd0/aLP2HMXzrqBthRdR7tSnRftbDw9ipNGfr0lxAyR1Zskse8Sg04ff</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 46 Bill Batch Response Sample

5.7 Bill Pull Service

ONEIC-TASDEED may ‘pull’ bill data from the Biller site by sending the Bill Pull message. The Biller billing application must respond with the bill data of the requested bill category and service type.

Operation Name: BillPull

Called by: ONEIC-TASDEED





C o n 2 U s
Integration Specs

Request Parameters:

Element Name	Mandatory/Conditional/Optional
AcctInfo	M
BillingNo	M
BillNo	O
End of 'AcctInfo' container	
ServiceType	M
PayerInfo	O
IdType	C
Id	C
Nation	C
Name	O
Phone	O
Address	O
Email	O
JOEBPPSNo	C
End of 'PayerInfo' container	

Response Parameters:

Element Name	Mandatory/Conditional/Optional
RecCount	M
BillsRec	C



C o n 2 U s
Integration Specs

BillRec - repeatable	M
Result	C
ErrorCode	M
ErrorDesc	M
Severity	M
End of 'Result' container	
AcctInfo	M
BillingNo	M
BillNo	M
End of 'AcctInfo' container	
BillStatus	M
DueAmount	M
IssueDate	M
OpenDate	O
DueDate	M
ExpiryDate	O
CloseDate	O
ServiceType	M
BillType	O
PmtConst	O

C o n t a i n e r s
Integration Specs

AllowPart	M
Lower	M
Upper	M
End of 'PmtConst' container	
SubPmts	C
SubPmt - repeatable	M
Amount	M
SetBnkCode	M
AcctNo	M
End of 'SubPmt' container	
End of ' SubPmts' container	
End of 'BillRec' container	
End of 'BillsRec' container	

Request Sample:

C o n 2 U s

Integration Specs

```
<MFEP>
<MsgHeader>
  <TmStp>2013-12-25T10:52:50</TmStp>
  <TrsInf>
    <SdrCode>1</SdrCode>
    <RcvCode>3</RcvCode>
    <ReqTyp>BILPULRQ</ReqTyp>
  </TrsInf>
</MsgHeader>
<MsgBody>
  <AcctInfo>
    <BillingNo>1325646</BillingNo>
  </AcctInfo>
  <ServiceType>FixedLine</ServiceType>
</MsgBody>
<MsgFooter>
  <Security>
    <Signature>aaahkdkabl+-0/aLP2HMXzrqBthRdR7tSnRftbDw9ipNGfr0lxAyR1Zskse8Sg04cx</Signature>
  </Security>
</MsgFooter>
</MFEP>
```

Figure 47 Bill Pull Request Sample

Response Sample:

C o n 2 U s

Integration Specs

```

<MFEP>
  <MsgHeader>
    <TmStp>2015-01-18T19:27:16</TmStp>
    <GUID>8d5cec8e-214a-4f8a-b724-a5794a4436c1</GUID>
    <TrsInf>
      <SdrCode>3</SdrCode>
      <ResTyp>BILPULRS</ResTyp>
    </TrsInf>
    <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </MsgHeader>
  <MsgBody>
    <RecCount>1</RecCount>
    <BillsRec>
      <BillRec>
        <Result>
          <ErrorCode>0</ErrorCode>
          <ErrorDesc>Success</ErrorDesc>
          <Severity>Info</Severity>
        </Result>
        <AcctInfo>
          <BillingNo>1325646</BillingNo>
          <BillNo>1325646</BillNo>
        </AcctInfo>
        <BillStatus>BillNew</BillStatus>
        <DueAmount>50</DueAmount>
        <IssueDate>2015-01-18T19:27:17</IssueDate>
        <DueDate>2015-01-18T19:27:17</DueDate>
        <ServiceType>FixedLine</ServiceType>
        <PmtConst>
          <AllowPart>true</AllowPart>
          <Lower>11</Lower>
          <Upper>200</Upper>
        </PmtConst>
      </BillRec>
    </BillsRec>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>
        eTv4/Ajoc0ghjYn9igfaR50gSUwWQf8Le/pzp+j0YPQl3hjmZYDrZvtQjV7
        AK9//o34y8jSyIptUorz/D/NxAjH3Izi6x95ahRKBllKBmTgD3fzTv3pB7U8PMBBPg==
      </Signature>
    </Security>
  </MsgFooter>
</MFEP>

```

Figure 48 Bill Pull Response Sample

5.8 Bill Inquiry Service

Bank/PSP may query ONEIC-TASDEED for a bill and associated data using a Bill Inquiry message. This message is flexible to support a variety of bill scenarios. In all cases, however, the inquiry is bill centric; payment data may only be fetched in the context of the associated bill.

Operation Name: Inquire

Called by: Banks/PSPs

Request Parameters:

Element Name	Mandatory/Conditional/Optional
AcctInfo	M
BillingNo	M
BillNo	O
BillerCode	M
End of 'AcctInfo' container	
ServiceType	M
PayerInfo	M (Review section 6.12.27)
IdType	C
Id	C
Nation	C
Name	O
Phone	O

C o n 2 U s
Integration Specs

Address	O
Email	O
JOEBPPSNo	C
End of 'PayerInfo' container	
DateRange	O
StartDt	M
EndDt	M
End of 'DateRange' container	
IncPayments	O
IncPaidBills	O

Response Parameters:

Element Name	Mandatory/Conditional/Optional
InqRefNo	C
RecCount	M
BillsRec	C
BillRec - repeatable	M
Result	C
ErrorCode	M
ErrorDesc	M

C o n 2 U s
Integration Specs

Severity	M
End of 'Result' container	
AcctInfo	M
BillingNo	M
BillNo	M
BillerCode	M
End of 'AcctInfo' container	
BillStatus	M
DueAmount	M
FeesAmt	M
IssueDate	M
OpenDate	O
DueDate	M
ExpiryDate	O
CloseDate	O
ServiceType	M
BillType	O
PmtConst	O
AllowPart	M
Lower	M

C o n t a i n e r s
Integration Specs

Upper	M
End of 'PmtConst' container	
BillPmts	O
BillPmt - repeatable	M
JOEBPPSTrx	M
BankCode	M
DueAmount	M
PaidAmt	M
PmtStatus	M
ProcessDate	M
End of 'BillPmt' container	
End of 'BillPmts' container	
End of 'BillRec' container	
End of 'BillsRec' container	

Request Sample:



C o n 2 U s

Integration Specs

```
<MFEP>
  <MsgHeader>
    <TmStp>2013-12-26T10:53:00</TmStp>
    <TrsInf>
      <SdrCode>2</SdrCode>
      <RcvCode>1</RcvCode>
      <ReqTyp>BILINQRQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  <MsgBody>
    <AcctInfo>
      <BillingNo>1325646</BillingNo>
      <BillerCode>3</BillerCode>
    </AcctInfo>
    <ServiceType>FixedLine</ServiceType>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>MgBzlRPEqoqwPJ86YRws+QrTj2Me8e3C5zc8lxmo82CFM+H5YC1305Ilkx0pJolG1/
        CQgBiPStbh7bqs+tXk80klsDMj6Wjm+ZKWPZIcc7fs7LLAUT6e6+NaxDfzp6xD5YgkRSZNSygQirHGM</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 49 Bill Inquiry Request Sample

Response Sample:



Con 2 Us

Integration Specs

```

<MFEP>
  <MsgHeader>
    <TmStp>2013-12-26T10:53:15</TmStp>
    <GUID>efc13c15-164a-4bd6-b407-3a224dec3ebe</GUID>
    <TrsInf>
      <RcvCode>3</RcvCode>
      <ResTyp>BILINQRS</ResTyp>
    </TrsInf>
    <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </MsgHeader>
  <MsgBody>
    <RecCount>1</RecCount>
    <BillsRec>
      <BillRec>
        <AcctInfo>
          <BillingNo>1325646</BillingNo>
          <BillNo>1325646</BillNo>
          <BillerCode>13</BillerCode>
        </AcctInfo>
        <BillStatus>BillNew</BillStatus>
        <DueAmount>10</DueAmount>
        <FeesAmt>0</FeesAmt>
        <IssueDate>2013-12-25T08:52:59</IssueDate>
        <DueDate>2013-12-30T08:52:59</DueDate>
        <ServiceType>FixedLine</ServiceType>
        <PmtConst>
          <AllowPart>false</AllowPart>
          <Lower>10</Lower>
          <Upper>10</Upper>
        </PmtConst>
      </BillRec>
    </BillsRec>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>TvJ19Fw4hPfd0/YIH2Ex6fUE4W6kqhsHXUD60xvXqb8l83dZDCLsAkHEKa6p
      YKIdVkJsbz9S/UIcNkUN/20WCnPKa431o7+qVuiRIhGpSDNuCuQKnwtYVQ==</Signature>
    </Security>
  </MsgFooter>
</MFEP>

```

Figure 50 Bill Inquiry Response Sample

5.9 Bill Payment Service



Con 2 Us

Integration Specs

The Bill Payment service permits Bank/PSP to create new payment records in ONEIC-TASDEED. The service is intended to involve a set of validations on the received payment, if the validation process is successful, ONEIC-TASDEED will generate a payment transaction number 'JOEBPPSTrx' and return it in the response message to the Bank/PSP.

Bank/PSP should get the latest bill details first by sending bill inquiry request.

Operation Name: PayBill

Called by: Banks/PSPs

Request Parameters:

Element Name	Mandatory/Conditional/Optional
Transactions	M
TrxInf - repeatable	M
AcctInfo	M
BillingNo	M
BillNo	M
BillerCode	M
End of 'AcctInfo' container	
InqRefNo	M
BankTrxID	M
PmtStatus	M
DueAmt	M
PaidAmt	M
ProcessDate	M



C o n 2 U s
Integration Specs

AccessChannel	M
PaymentMethod	M
PaymentType	O
Currency	O
ServiceTypeDetails	M
ServiceType	M
End of 'ServiceTypeDetails' container	
PayerInfo	M (Review section 6.13.32)
IdType	C
Id	C
Nation	C
Name	O
Phone	O
Address	O
Email	O
JOEBPPSNo	C
End of 'PayerInfo' container	
End of 'TrxInf' container	
End of 'Transactions' container	

Response Parameters:

Element Name	Mandatory/Conditional/Optional
Transactions	M
TrxInf - repeatable	M
AcctInfo	M
BillingNo	M
BillNo	M
BillerCode	M
End of 'AcctInfo' container	
BankTrxID	M
PmtStatus	M
DueAmt	M
PaidAmt	M
ProcessDate	M
STMTDate	C
JOEBPPSTrx	C
Result	M
ErrorCode	M
ErrorDesc	M
Severity	M

End of 'Result' container
End of 'TrxInf' container
End of 'Transactions' container

Request Sample:

```

<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T12:53:00</TmStp>
    <TrsInf>
      <SdrCode>2</SdrCode>
      <RcvCode>1</RcvCode>
      <ReqTyp>BILPMTRQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  <MsgBody>
    <Transactions>
      <TrxInf>
        <AcctInfo>
          <BillingNo>1325646</BillingNo>
          <BillNo>1325646</BillNo>
          <BillerCode>3</BillerCode>
        </AcctInfo>
        <BankTrxID>RAJ32165489</BankTrxID>
        <PmtStatus>PmtNew</PmtStatus>
        <DueAmt>10</DueAmt>
        <PaidAmt>10</PaidAmt>
        <ProcessDate>2013-12-25T12:53:00</ProcessDate>
        <AccessChannel>ATM</AccessChannel>
        <PaymentMethod>CCARD</PaymentMethod>
        <ServiceTypeDetails>
          <ServiceType>FixedLine</ServiceType>
        </ServiceTypeDetails>
      </TrxInf>
    </Transactions>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>abchkdkabl++0/aLP2HMXzrqBthRdR7tSnRFtbDw9ipNGfr0lxAyRlZskse8Sg0as</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 51 Bill Payment Request Sample

Response Sample:

C o n 2 U s

Integration Specs

```

<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T12:53:20</TmStp>
    <GUID>c2967db5-25c5-4d71-9a85-f82f0649ded3</GUID>
    <TrsInf>
      <RcvCode>2</RcvCode>
      <ResTyp>BILPMTRS</ResTyp>
    </TrsInf>
    <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </MsgHeader>
  <MsgBody>
    <Transactions>
      <TrxInf>
        <AcctInfo>
          <BillingNo>1325646</BillingNo>
          <BillNo>1325646</BillNo>
          <BillerCode>3</BillerCode>
        </AcctInfo>
        <BankTrxID>RAJ32165489</BankTrxID>
        <PmtStatus>PmtNew</PmtStatus>
        <DueAmt>10</DueAmt>
        <PaidAmt>10</PaidAmt>
        <ProcessDate>2013-12-25T12:53:00</ProcessDate>
        <STMTDate>2013-12-25</STMTDate>
        <JOEBPPSTrx>1509</JOEBPPSTrx>
        <Result>
          <ErrorCode>0</ErrorCode>
          <ErrorDesc>Success</ErrorDesc>
          <Severity>Info</Severity>
        </Result>
      </TrxInf>
    </Transactions>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>BthRdR7tSnRftbDw9ipNGfr0lxAyR1zskse8Sg022+Ixm0WpQLabPus6Ml8qIPY8g9c23</Signature>
    </Security>
  </MsgFooter>
</MFEP>

```

Figure 52 Bill Payment Response Sample

5.10 Prepaid Validation Service

The Bank/PSP Prepaid Validation service permits Bank/PSP to initiate a prepaid transaction to validate it before payment by ONEIC-TASDEED. The service is intended to involve a set of validations on the potential prepaid payment, if the validation process is successful, ONEIC-



Con 2 Us

Integration Specs

TASDEED will generate and send a validation code to the Bank/PSP in the response message. Before sending the prepaid payment transaction, Bank/PSP should validate the prepaid transaction first using the prepaid validation request.

Operation Name: Validate

Called by: Banks/PSPs

Request Parameters:

Element Name	Mandatory/Conditional/Optional
BillingInfo	M
AcctInfo	M
BillingNo	C
BillerCode	M
End of 'AcctInfo' container	
DueAmt	C
ServiceTypeDetails	M
ServiceType	M
PrepaidCat	C
End of 'ServiceTypeDetails' container	
PayerInfo	M (Review section 6.14.14)
IdType	C
Id	C
Nation	C
Name	O



C o n 2 U s

Integration Specs

Phone	O
Address	O
Email	O
JOEBPPSNo	C
End of 'PayerInfo' container	
End of 'BillingInfo' container	

Response Parameters:

Element Name	Mandatory/Conditional/Optional
BillingInfo	M
Result	M
ErrorCode	M
ErrorDesc	M
Severity	M
End of 'Result' container	
AcctInfo	C
BillingNo	C
BillerCode	M
End of 'AcctInfo' container	
DueAmt	C
FeesAmt	C
ValidationCode	C
MsgLabel	O
ServiceTypeDetails	C
ServiceType	M
PrepaidCat	C
End of 'ServiceTypeDetails' container	
End of 'BillingInfo' container	

Request Sample:

```
<MFEP>
  <MsgHeader>
    <TmStp>2014-06-19T06:53:00</TmStp>
    <GUID>8b4b4070-2045-46f2-83e5-e38be49ed7d8</GUID>
    <TrsInf>
      <SdrCode>77</SdrCode>
      <RcvCode>1</RcvCode>
      <ReqTyp>PREPADVALRQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  <MsgBody>
    <BillingInfo>
      <AcctInfo>
        <BillingNo>07xxxxxxxx</BillingNo>
        <BillerCode>42</BillerCode>
      </AcctInfo>
      <ServiceTypeDetails>
        <ServiceType>Mobile</ServiceType>
        <PrepaidCat>JD_5</PrepaidCat>
      </ServiceTypeDetails>
    </BillingInfo>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>12chdkabl+-0/aLP2HMXzrqBthRdR7tSnRFtbDw9dkabl+-0/aLP2HMXzrqBtqBthRdR7tSnRFtbDw9dkabl+</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 53 Bank/PSP Prepaid Validation Request Sample

Response Sample:



C o n 2 U s

Integration Specs

```
<MFEP>
  <MsgHeader>
    <TmStp>2014-06-19T06:53:02</TmStp>
    <GUID>8b4b4070-2045-46f2-83e5-e38be49ed7d8</GUID>
    <TrsInf>
      <RcvCode>77</RcvCode>
      <ResTyp>PREPADVALRS</ResTyp>
    </TrsInf>
    <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </MsgHeader>
  <MsgBody>
    <BillingInfo>
      <Result>
        <ErrorCode>0</ErrorCode>
        <ErrorDesc>Success</ErrorDesc>
        <Severity>Info</Severity>
      </Result>
      <AcctInfo>
        <BillingNo>07xxxxxxxxx</BillingNo>
        <BillerCode>42</BillerCode>
      </AcctInfo>
      <DueAmt>7.5</DueAmt>
      <ValidationCode>50421</ValidationCode>
      <MsgLabel>Sample Message</MsgLabel>
      <ServiceTypeDetails>
        <ServiceType>Mobile</ServiceType>
        <PrepaidCat>JD_5</PrepaidCat>
      </ServiceTypeDetails>
    </BillingInfo>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>asdxckdkabl+-0/aLP2HMXzrqBthRdR7tSnRFtbDw9dkabl4sddw2HMXzrqBtqBthRdR7tSnRFtbDw9dkaff</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 54 Bank/PSP Prepaid Validation Response Sample

5.11 Biller Prepaid Validation Service

The Biller Prepaid Validation service is used to ask the biller to validate a prepaid transaction before giving the approval to the Bank/PSP to send a payment for it. ONEIC-TASDEED will initiate the request for the prepaid services that the Biller required to validate it, and the Biller





Con 2 Us

Integration Specs

system must respond on a real-time with the required data of the requested transaction that holds a specific validation code.

Operation Name: PrepaidValidation

Called by: ONEIC-TASDEED

Request Parameters:

Element Name	Mandatory/Conditional/Optional
BillingInfo	M
AcctInfo	M
BillingNo	C
BillerCode	M
End of 'AcctInfo' container	
DueAmt	C
ValidationCode	M
ServiceTypeDetails	M
ServiceType	M
PrepaidCat	C
End of 'ServiceTypeDetails' container	
PayerInfo	O
IdType	C
Id	C
Nation	C





C o n 2 U s

Integration Specs

Name	O
Phone	O
Address	O
Email	O
JOEBPPSNo	C
End of 'PayerInfo' container	
End of 'BillingInfo' container	

Response Parameters:

Element Name	Mandatory/Conditional/Optional
BillingInfo	M
Result	M
ErrorCode	M
ErrorDesc	M
Severity	M
End of 'Result' container	
AcctInfo	C
BillingNo	C
BillerCode	M
End of 'AcctInfo' container	



C o n 2 U s
Integration Specs

DueAmt	C
ValidationCode	C
MsgLabel	O
ServiceTypeDetails	C
ServiceType	M
PrepaidCat	C
End of 'ServiceTypeDetails' container	
End of 'BillingInfo' container	

Request Sample:

```
- <MFEP xmlns="">
  - <MsgHeader>
    <TmStp>2015-12-17T16:51:55</TmStp>
    <GUID>1d06359d-35bd-4800-a51e-c73512050a12</GUID>
  - <TrsInf>
    <RcvCode>116</RcvCode>
    <ReqTyp>BILRPREPADVALRQ</ReqTyp>
  </TrsInf>
</MsgHeader>
- <MsgBody>
  - <BillingInfo>
    - <AcctInfo>
      <BillingNo>12345555678</BillingNo>
      <BillerCode>116</BillerCode>
    </AcctInfo>
    <DueAmt>10</DueAmt>
    <ValidationCode>35112</ValidationCode>
  - <ServiceTypeDetails>
    <ServiceType>Internet</ServiceType>
  </ServiceTypeDetails>
</BillingInfo>
</MsgBody>
- <MsgFooter>
  - <Security>
    <Signature>XLimXSia/uLidCcQfKIWeD1mYFo9pKuZpkPhMVfkq/rdHwhG5bLOIj0J4qQahRkN9w==</Signature>
  </Security>
</MsgFooter>
</MFEP>
```

Figure 55 Biller Prepaid Validation Request Sample

Response Sample:

C o n 2 U s

Integration Specs

```

- <MFEP xmlns="">
  - <MsgHeader>
    <TmStp>2015-12-17T16:51:57</TmStp>
    <GUID>1d06359d-35bd-4800-a51e-c73512050a12</GUID>
  - <TrsInf>
    <SdrCode>116</SdrCode>
    <ResTyp>BILRPREPADVALRS</ResTyp>
  </TrsInf>
  - <Result>
    <ErrorCode>0</ErrorCode>
    <ErrorDesc>Success</ErrorDesc>
    <Severity>Info</Severity>
  </Result>
</MsgHeader>
- <MsgBody>
  - <BillingInfo>
    - <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
    - <AcctInfo>
      <BillingNo>12345555678</BillingNo>
      <BillerCode>116</BillerCode>
    </AcctInfo>
    <DueAmt>10</DueAmt>
    <ValidationCode>35112</ValidationCode>
    - <ServiceTypeDetails>
      <ServiceType>Internet</ServiceType>
    </ServiceTypeDetails>
  </BillingInfo>
</MsgBody>
- <MsgFooter>
  - <Security>
    <Signature>PimXu4/dpNxsXLhoCcQfKIWeD1mYFo9pKuZpkPhMXUvsVfj/HwhG5bL81w==</Signature>
  </Security>
</MsgFooter>
</MFEP>

```

Figure 56 Biller Prepaid Validation Response Sample

5.12 Prepaid Payment Service

The Prepaid Payment service permits Bank/PSP to create a new payment record in ONEIC-TASDEED. The service is intended to involve a set of validations on the received payment, if the validation process is successful, ONEIC-TASDEED will generate a payment transaction number 'JOEBPPSTrx' for the validation code of that payment request and return it in the response message to the Bank/PSP. Bank/PSP should validate the prepaid transaction first before sending the payment request.

Operation Name: Pay

Called by: Banks/PSPs

Request Parameters:

Element Name	Mandatory/Conditional/Optional
TrxInf	M
AcctInfo	M
BillingNo	C
BillerCode	M
End of 'AcctInfo' container	
BankTrxID	M
ValidationCode	M
PmtStatus	M
DueAmt	M
PaidAmt	M
ProcessDate	M
AccessChannel	M
PaymentMethod	M
PaymentType	O
Currency	O
ServiceTypeDetails	M
ServiceType	M

C o n 2 U s
Integration Specs

PrepaidCat	C
End of 'ServiceTypeDetails' container	
PayerInfo	M (Review section 6.16.20)
IdType	C
Id	C
Nation	C
Name	O
Phone	O
Address	O
Email	O
JOEBPPSNo	C
End of 'PayerInfo' container	
End of 'TrxInf' container	

Response Parameters:

Element Name	Mandatory/Conditional/Optional
TrxInf	M
AcctInfo	M
BillingNo	C



C o n t a i n e r s
Integration Specs

BillerCode	M
End of 'AcctInfo' container	
BankTrxID	M
ValidationCode	M
PmtStatus	M
DueAmt	M
PaidAmt	M
ProcessDate	M
STMTDate	C
JOEBPPSTrx	C
Result	M
ErrorCode	M
ErrorDesc	M
Severity	M
End of 'Result' container	
End of 'TrxInf' container	

Request Sample:



C o n 2 U s

Integration Specs

```

<MFEP>
  <MsgHeader>
    <TmStp>2014-06-19T06:53:02</TmStp>
    <GUID>2a4b4070-2045-46f2-83e5-e38be49ed71c</GUID>
    <TrsInf>
      <SdrCode>77</SdrCode>
      <RcvCode>1</RcvCode>
      <ReqTyp>PREPADPMTRQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  <MsgBody>
    <TrxInf>
      <AcctInfo>
        <BillingNo>07xxxxxxxx</BillingNo>
        <BillerCode>42</BillerCode>
      </AcctInfo>
      <BankTrxID>RAJ32165489</BankTrxID>
      <ValidationCode>50421</ValidationCode>
      <PmtStatus>PmtNew</PmtStatus>
      <DueAmt>7.5</DueAmt>
      <PaidAmt>7.5</PaidAmt>
      <ProcessDate>2014-06-19T06:53:02</ProcessDate>
      <AccessChannel>ATM</AccessChannel>
      <PaymentMethod>CCARD</PaymentMethod>
      <ServiceTypeDetails>
        <ServiceType>Mobile</ServiceType>
        <PrepaidCat>JD_5</PrepaidCat>
      </ServiceTypeDetails>
    </TrxInf>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>acdkdkabl+-0/aLP2HMXzrqBthRdR7tSnRFtbDw9dkab14sddw2HMXzrqBtqBthRdR7tSnRFtbDw9dka22</Signature>
    </Security>
  </MsgFooter>
</MFEP>

```

Figure 57 Prepaid Payment Request Sample

Response Sample:

C o n 2 U s

Integration Specs

```
<MFEP>
  <MsgHeader>
    <TmStp>2014-06-19T06:53:02</TmStp>
    <GUID>334b4070-d045-46f2-83e5-e38be49ed71f</GUID>
    <TrsInf>
      <RcvCode>77</RcvCode>
      <ResTyp>PREPADPMTRS</ResTyp>
    </TrsInf>
    <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </MsgHeader>
  <MsgBody>
    <TrxInf>
      <AcctInfo>
        <BillingNo>07xxxxxxxx</BillingNo>
        <BillerCode>42</BillerCode>
      </AcctInfo>
      <BankTrxID>RAJ32165489</BankTrxID>
      <ValidationCode>50421</ValidationCode>
      <PmtStatus>PmtNew</PmtStatus>
      <DueAmt>7.5</DueAmt>
      <PaidAmt>7.5</PaidAmt>
      <ProcessDate>2014-06-19T06:53:02</ProcessDate>
      <STMTDate>2014-06-19</STMTDate>
      <JOEBPPSTrx>1509999</JOEBPPSTrx>
    <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </TrxInf>
</MsgBody>
<MsgFooter>
  <Security>
    <Signature>BthRdR7tSnRFtbDw9ipNGfr0lxAyRlZskse8Sg022+Ixm0WpQLabPus6Ml8qIPY8g9c23</Signature>
  </Security>
</MsgFooter>
</MFEP>
```

Figure 58 Prepaid Payment Response Sample

5.13 Payment Notification Service

Payment Notification service is used to alert Biller and Biller's Settlement Bank of the funds collected on a given date by the 'Paying Bank'/'PSP'. These notifications may be sent either as real-time notifications or as batch file in offline mode.

Payment notification process divided into two-separated services:

1. Biller Payment Notification

- **Operation** **Name:**

ReceivePaymentNotification

2. Settlement Bank Payment Notification

- **Operation** **Name:**

ReceivePaymentNotification **Called by:** ONEIC-TASDEED

Request Parameters:

Element Name	Mandatory/Conditional/Optional
Transactions	M
TrxInf - repeatable	M
AcctInfo	M
BillingNo	C
BillNo	M
BillerCode	C
End of 'AcctInfo' container	
JOEBPPSTrx	M
BankTrxID	M
PmtSrc	O

C o n 2 U s
 Integration Specs

BankCode	M
PmtStatus	M
DueAmt	M
PaidAmt	M
FeesAmt	M
FeesOnBiller	M
ProcessDate	M
STMTDate	M
AccessChannel	M
PaymentMethod	M
PaymentType	O
Currency	O
ServiceTypeDetails	M
ServiceType	M
PrepaidCat	C
End of 'ServiceTypeDetails' container	
SubPmts	C
SubPmt – repeatable	M
Amount	M



C o n 2 U s
Integration Specs

SetBnkCode	M
AcctNo	M
End of 'SubPmt' container	
End of ' SubPmts' container	
PayerInfo	O
IdType	C
Id	C
Nation	C
Name	O
Phone	O
Address	O
Email	O
JOEBPPSNo	C
End of 'PayerInfo' container	
End of 'TrxInf' container	
End of 'Transactions' container	

Response Parameters:

Element Name	Mandatory/Conditional/Optional
Transactions	M



C o n 2 U s
Integration Specs

TrxInf - repeatable	M
JOEBPPSTrx	M
ProcessDate	M
STMTDate	M
Result	M
ErrorCode	M
ErrorDesc	M
Severity	M
End of 'Result' container	
End of 'TrxInf' container	
End of 'Transactions' container	

Request Sample for Bank:

```

<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T12:55:00</TmStp>
    <TrsInf>
      <SdrCode>1</SdrCode>
      <RcvCode>9</RcvCode>
      <ReqTyp>BNKPMTNTFRQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  <MsgBody>
    <Transactions>
      <TrxInf>
        <AcctInfo>
          <BillingNo>1325646</BillingNo>
          <BillNo>1325646</BillNo>
          <BillerCode>3</BillerCode>
        </AcctInfo>
        <JOEBPPSTrx>1509</JOEBPPSTrx>
        <BankTrxID>Raj12658</BankTrxID>
        <BankCode>3</BankCode>
        <PmtStatus>PmtNew</PmtStatus>
        <DueAmt>10</DueAmt>
        <PaidAmt>10</PaidAmt>
        <FeesAmt>1</FeesAmt>
        <FeesOnBiller>true</FeesOnBiller>
        <ProcessDate>2013-12-25T12:53:00</ProcessDate>
        <STMTDate>2013-12-25</STMTDate>
        <AccessChannel>ATM</AccessChannel>
        <PaymentMethod>CCARD</PaymentMethod>
        <ServiceTypeDetails>
          <ServiceType>FixedLine</ServiceType>
        </ServiceTypeDetails>
      </TrxInf>
    </Transactions>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>126hkdkvel++0/aLP2HMXzrqBthRdR7tSnRftbDw9ipNGfr0lxAyRlZskse+Ixm0Wp</Signature>
    </Security>
  </MsgFooter>
</MFEP>

```

Figure 59 Bank Payment Notification Request Sample

Response Sample for Bank:



C o n 2 U s

Integration Specs

```
<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T12:55:00</TmStp>
    <GUID>15c72631-60d1-4686-b552-9517ff0335d5</GUID>
    <TrsInf>
      <SdrCode>9</SdrCode>
      <ResTyp>BNKPMTNTFRS</ResTyp>
    </TrsInf>
    <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </MsgHeader>
  <MsgBody>
    <Transactions>
      <TrxInf>
        <JOEBPPSTrx>1509</JOEBPPSTrx>
        <ProcessDate>2013-12-25T12:53:00</ProcessDate>
        <STMTDate>2013-12-25</STMTDate>
        <Result>
          <ErrorCode>0</ErrorCode>
          <ErrorDesc>Success</ErrorDesc>
          <Severity>Info</Severity>
        </Result>
      </TrxInf>
    </Transactions>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>aLP2HMXzrqBthRdR7tSnRftbDw9ipNGfr0lxAyR1Zskse85g0+++Ixm0WpQLZ6Pus6M</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 60 Bank Payment Notification Response Sample

Request Sample for Biller:



Con 2 Us

Integration Specs

```

<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T12:55:00</TmStp>
    <TrsInf>
      <SdrCode>1</SdrCode>
      <RcvCode>3</RcvCode>
      <ReqTyp>BLRPMTNTFRQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  <MsgBody>
    <Transactions>
      <TrxInf>
        <AcctInfo>
          <BillingNo>1325646</BillingNo>
          <BillNo>1325646</BillNo>
        </AcctInfo>
        <JOEBPPSTrx>1509</JOEBPPSTrx>
        <BankTrxID>Raj12658</BankTrxID>
        <BankCode>3</BankCode>
        <PmtStatus>PmtNew</PmtStatus>
        <DueAmt>10</DueAmt>
        <PaidAmt>10</PaidAmt>
        <FeesAmt>1</FeesAmt>
        <FeesOnBiller>true</FeesOnBiller>
        <ProcessDate>2013-12-25T12:53:00</ProcessDate>
        <STMTDate>2013-12-25</STMTDate>
        <AccessChannel>ATM</AccessChannel>
        <PaymentMethod>CCARD</PaymentMethod>
        <ServiceTypeDetails>
          <ServiceType>FixedLine</ServiceType>
        </ServiceTypeDetails>
      </TrxInf>
    </Transactions>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>126hkdkvel++0/aLP2HMXzrqBthRdR7tSnRFtbDw9ipNGfr0lxAyRlZskse+Ixm0</Signature>
    </Security>
  </MsgFooter>
</MFEP>

```

Figure 61 Biller Payment Notification Request Sample

Response Sample for Biller:



Con 2 Us

Integration Specs

```
<MFEP>
  <MsgHeader>
    <TmStp>2013-12-25T12:55:10</TmStp>
    <GUID>15c72631-60d1-4686-b552-9517ff0335d5</GUID>
    <TrsInf>
      <SdrCode>3</SdrCode>
      <ResTyp>BLRPMTNTFRS</ResTyp>
    </TrsInf>
    <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </MsgHeader>
  <MsgBody>
    <Transactions>
      <TrxInf>
        <JOEBPPSTrx>1509</JOEBPPSTrx>
        <ProcessDate>2013-12-25T12:53:00</ProcessDate>
        <STMTDate>2013-12-25</STMTDate>
        <Result>
          <ErrorCode>0</ErrorCode>
          <ErrorDesc>Success</ErrorDesc>
          <Severity>Info</Severity>
        </Result>
      </TrxInf>
    </Transactions>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>aLP2HMXzrqBthRdR7tSnRFtbDw9ipNGfr0lxAyRlZskse8Sg0+++Ixm0WpQ</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 62 Biller Payment Notification Response Sample

5.14 Payment Acknowledgment Service

Payment Acknowledgment service is used to alert the 'PSP Settlement Banks' about the successful payments collected by its PSP(s) on a specific Settlement Date. Also, it can be sent for 'PSPs' and 'Paying Banks' to provide them with more details about their successful payments.

These alerts/notifications may be sent either online as real-time notifications or offline by downloading files from the Standalone Application.



Operation Name: ReceivePaymentAcknowledgment

Called by: ONEIC-TASDEED

Request Parameters:

Element Name	Mandatory/Conditional/Optional
Transactions	M
TrxInf - repeatable	M
AcctInfo	M
BillingNo	C
BillNo	C
BillerCode	M
End of 'AcctInfo' container	
JOEBPPSTrx	M
ParTrxID	M
ValidationCode	C
PSPCode	C

PmtSrc	M
PmtStatus	M
DueAmt	M
PaidAmt	M

C o n t a i n e r s
Integration Specs

FeesAmt	C
ParFees	M
FeesOnBiller	M
ProcessDate	M
STMTDate	M
AccessChannel	M
PaymentMethod	M
PaymentType	M
Currency	O
ServiceTypeDetails	M
ServiceType	M
PrepaidCat	C
End of 'ServiceTypeDetails' container	
PayerInfo	O
IdType	C
Id	C
Nation	C
Name	O
Phone	O
Address	O

C o n 2 U s

Integration Specs

Email	O
JOEBPPSNo	C
End of 'PayerInfo' container	
End of 'TrxInf' container	
End of 'Transactions' container	

Response Parameters:

Element Name	Mandatory/Conditional/Optional
Transactions	M
TrxInf - repeatable	M
JOEBPPSTrx	M
ProcessDate	M
STMTDate	M
Result	M
ErrorCode	M
ErrorDesc	M
Severity	M
End of 'Result' container	
End of 'TrxInf' container	
End of 'Transactions' container	

Request Sample:



C o n 2 U s

Integration Specs

```
<MFEP>
  <MsgHeader>
    <TmStp>2014-10-16T11:09:26</TmStp>
    <TrsInf>
      <SdrCode>1</SdrCode>
      <RcvCode>2</RcvCode>
      <ReqTyp>PMTACKRQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  <MsgBody>
    <Transactions>
      <TrxInf>
        <AcctInfo>
          <BillingNo>9182731827</BillingNo>
          <BillNo>9182731827</BillNo>
          <BillerCode>5</BillerCode>
        </AcctInfo>
        <JOEBPPSTrx>201410162610</JOEBPPSTrx>
        <ParTrxID>AP798755</ParTrxID>
        <PmtSrc>BNKPmt</PmtSrc>
        <PmtStatus>PmtNew</PmtStatus>
        <DueAmt>112.100</DueAmt>
        <PaidAmt>112.100</PaidAmt>
        <FeesAmt>12.1</FeesAmt>
        <ParFees>0.264</ParFees>
        <FeesOnBiller>false</FeesOnBiller>
        <ProcessDate>2014-10-16T11:09:26</ProcessDate>
        <STMTDate>2014-10-18</STMTDate>
        <AccessChannel>ATM</AccessChannel>
        <PaymentMethod>CCARD</PaymentMethod>
        <PaymentType>Postpaid</PaymentType>
        <ServiceTypeDetails>
          <ServiceType>mobile</ServiceType>
        </ServiceTypeDetails>
      </TrxInf>
    </Transactions>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>12chkdkabl+-0/aLP2HMXzrqBthRdR7tSnR12chkdkabl+-
      0/aLP2HMXzFtbDw9BthRdR7tSnRtSnR12chkdkaw9BthRdR12ch</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 63 Payment Acknowledgment Request Sample

Response Sample:





Con 2 Us

Integration Specs

```
<MFEP>
  <MsgHeader>
    <TmStp>2014-10-16T11:09:27</TmStp>
    <GUID>15c72631-60d1-4686-b552-9517ff0335d5</GUID>
    <TrsInf>
      <SdrCode>2</SdrCode>
      <ResTyp>PMTACKRS</ResTyp>
    </TrsInf>
    <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </MsgHeader>
  <MsgBody>
    <Transactions>
      <TrxInf>
        <JOEBPPSTrx>201410162610</JOEBPPSTrx>
        <ProcessDate>2014-10-16T11:09:26</ProcessDate>
        <STMTDate>2014-10-18</STMTDate>
        <Result>
          <ErrorCode>0</ErrorCode>
          <ErrorDesc>Success</ErrorDesc>
          <Severity>Info</Severity>
        </Result>
      </TrxInf>
    </Transactions>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>aLP2HMXzrqBthRdR7tSnRFtbDw9ipNGfr0lxAyRlZskse8Sg0+++Ixm0WpQ</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 64 Payment Acknowledgment Response Sample

5.15 Payment Inquiry Service

Banks/PSPs may query ONEIC-TASDEED for a payment data using a Payment Inquiry service. However, the inquiry is payment centric; payment data may only be fetched in the context in case of ONEIC-TASDEED received the bill or prepaid payment request.

Operation Name: Inquire





C o n 2 U s
Integration Specs

Called by: Banks/PSPs

Request Parameters:

Element Name	Mandatory/Conditional/Optional
Transactions	M
TrxInf - repeatable	M
PmtGuid	M
AcctInfo	M
BillingNo	C
BillNo	C
BillerCode	M
End of 'AcctInfo' container	
ParTrxID	M
ValidationCode	C
DueAmt	M
PaidAmt	M
ProcessDate	M
AccessChannel	M
PaymentMethod	M
PaymentType	M
ServiceTypeDetails	M



C o n 2 U s
Integration Specs

ServiceType	M
PrepaidCat	C
End of 'ServiceTypeDetails' container	
End of 'TrxInf' container	
End of 'Transactions' container	

Response Parameters:

Element Name	Mandatory/Conditional/Optional
RecCount	M
Transactions	M
TrxInf - repeatable	M
PmtGuid	M
AcctInfo	M
BillingNo	C
BillNo	C
BillerCode	M
End of 'AcctInfo' container	
Validation Code	C
ParTrxID	M
PmtStatus	C

C o n t a i n e r s
Integration Specs

DueAmt	M
PaidAmt	M
ParFees	C
ProcessDate	M
STMTDate	C
JOEBPPSTrx	C
Access Channel	M
Payment Method	M
Payment Type	M
ServiceTypeDetails	M
ServiceType	M
Prepaid Cat	C
End of 'ServiceTypeDetails' container	
Result	M
ErrorCode	M
ErrorDesc	M
Severity	M
End of 'Result' container	
End of 'TrxInf' container	
End of 'Transactions' container	

Request Sample:

```
<MFEP>
  <MsgHeader>
    <TmStp>2014-11-03T15:08:53</TmStp>
    <TrsInf>
      <SdrCode>9</SdrCode>
      <RcvCode>1</RcvCode>
      <ReqTyp>PMTINQRQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  <MsgBody>
    <Transactions>
      <TrxInf>
        <PmtGuid>65a7c882-6315-4716-bb60-99d136023cb6</PmtGuid>
        <AcctInfo>
          <BillingNo>3</BillingNo>
          <BillNo>3</BillNo>
          <BillerCode>5</BillerCode>
        </AcctInfo>
        <ParTrxID>AK546138</ParTrxID>
        <DueAmt>110.00</DueAmt>
        <PaidAmt>110.00</PaidAmt>
        <ProcessDate>2014-11-03T09:54:11</ProcessDate>
        <AccessChannel>ATM</AccessChannel>
        <PaymentMethod>CCARD</PaymentMethod>
        <PaymentType>Postpaid</PaymentType>
        <ServiceTypeDetails>
          <ServiceType>Electricity</ServiceType>
        </ServiceTypeDetails>
      </TrxInf>
    </Transactions>
  </MsgBody>
  <MsgFooter>
    <Security>
      <Signature>3FWSxbTKXbLrPaBhCe56X6+yNugBVdKYkJ0K5+i7IMkNmFw3L7dE08gC8feQ7SCOII79N5AIe1XA
        Y83i7PtVe1yMK6Tpdkj7gwb1/fpqXLP4c8uWzLXppudIq0ISSIAttqUmgGzMW2XrQ+Tat2w8bjw==</Signature>
    </Security>
  </MsgFooter>
</MFEP>
```

Figure 65 Payment Inquiry Request Sample

Response Sample:

Con 2 Us

Integration Specs

```

<MFEP>
  <MsgHeader>
    <TmStp>2014-11-03T15:09:55</TmStp>
    <GUID>046aec78-d780-40d1-a959-a21a232e102f</GUID>
    <TrsInf>
      <RcvCode>9</RcvCode>
      <ResTyp>PMTINQRS</ResTyp>
    </TrsInf>
    <Result>
      <ErrorCode>0</ErrorCode>
      <ErrorDesc>Success</ErrorDesc>
      <Severity>Info</Severity>
    </Result>
  </MsgHeader>
  <MsgBody>
    <RecCount>1</RecCount>
    <Transactions>
      <TrxInf>
        <PmtGuid>65a7c882-6315-4716-bb60-99d136023cb6</PmtGuid>
        <AcctInfo>
          <BillingNo>3</BillingNo>
          <BillNo>3</BillNo>
          <BillerCode>5</BillerCode>
        </AcctInfo>
        <ParTrxID>AK546138</ParTrxID>
        <PmtStatus>PmtNew</PmtStatus>
        <DueAmt>110.00</DueAmt>
        <PaidAmt>110.00</PaidAmt>
        <ParFees>0.2600</ParFees>
        <ProcessDate>2014-11-03T09:54:11</ProcessDate>
        <STMTDate>2014-11-04</STMTDate>
        <JOEBPPSTrx>201411039865</JOEBPPSTrx>
        <AccessChannel>ATM</AccessChannel>
        <PaymentMethod>CCARD</PaymentMethod>
        <PaymentType>Postpaid</PaymentType>
        <ServiceTypeDetails>
          <ServiceType>Electricity</ServiceType>
        </ServiceTypeDetails>
      <Result>
        <ErrorCode>0</ErrorCode>
        <ErrorDesc>Success</ErrorDesc>
        <Severity>Info</Severity>
      </Result>
    </TrxInf>
  </Transactions>
</MsgBody>
<MsgFooter>
  <Security>xnbTKXbLrPaBhCe56X6+yNugBVdKYkJ0K5+i7IMkNnmFw3L7dE08gC8feQ7SCOII79N5AIe1XA
  Y83i7PtVe1yMK6Tpdbk7gwb1/fpqXLP4c8uWzLXppudIq0ISSIAttqUmgGzMW2XrQ+Tat2w8b</Security>
</MsgFooter>
</MFEP>

```

Figure 66 Payment Inquiry Response Sample

5.16 Biller Service Availability

ONEIC-TASDEED health service (Biller Service Availability) , it's a scheduled service that sends a scheduled heartbeat request to the billing system (Biller System) through Bill Pull and Biller Prepaid Validation service to verify that it is responding. When a specified number of heartbeats fail to arrive or there is an issue with the biller billing system, the service will send a notification message (As shown below) to all subscribed Banks/PSPs notifying them that the biller service is not available therefore preventing any inquiry or payment requests to be initiated on this biller.

Operation Name: RecvBillerSvcAvailabilityStuts

Called by: ONEIC-TASDEED

Request Parameters:

Element Name	Mandatory/Conditional/Optional
Biller	M
Code	M
Status	M
Services	C
Service	M
Code	M
Type	M
PaymentType	M
Status	M
End of 'Service' container	
End of 'Services' container	

End of 'Biller' container
Response Parameters:

The Banks/PSPs will return a Boolean value (True, False), True in case of received the notification successfully, otherwise will return a false.

Request Sample:

```

<MFEP>
  <MsgHeader>
    <TmStp>2016-04-10T11:09:24</TmStp>
    <TrsInf>
      <SdrCode>1</SdrCode>
      <RcvCode>21</RcvCode>
      <ReqTyp>BLRSVCAVARQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  <MsgBody>
    <Biller>
      <Code>9</Code>
      <Status>Up</Status>
      <Services>
        <Service>
          <Code>10</Code>
          <Type>3G</Type>
          <PaymentType>Prepaid</PaymentType>
          <Status>Down</Status>
        </Service>
        <Service>
          <Code>11</Code>
          <Type>Mobile</Type>
          <PaymentType>Postpaid</PaymentType>
          <Status>Up</Status>
        </Service>
      </Services>
    </Biller>
  </MsgBody>
  <MsgFooter>
    <Signature>BeQIGSRXLi8w3YiGFYNaxa3oR0hNKsLuA4QLH20EM4VmR4hGGZ74NEkQ
    +Z+CJrlKXrJUGO+oCM0ubyQLCnTz0Gj8P7qUILXuK+QhSqNgdeM2iTThg55582hJPhy</Signature>
  </MsgFooter>
</MFEP>
```

Figure

67 Biller Service Availability Request Sample

Response Sample:
True/False

5.17 System Service Availability

ONEIC-TASDEED health service (System Service Availability), this service will be responsible to sends a broadcast notification for all participants (Billers/Banks/PSPs) to notify them with the current status of ONEIC-TASDEED system to take the needed action from their side.

Operation Name: RecvSysSvcAvailabilityStuts

Called by: ONEIC-TASDEED

Request Parameters:

Element Name	Mandatory/Conditional/Optional
SysAval	M
Status	M
Period	O
From	M
To	M
End of 'Period' container	
Desc	O
End of 'SysAval' container	

Response Parameters:

The participants (Billers/Banks/PSPs) will return a Boolean value (True, False), True in case of received the notification successfully, otherwise will return a false.

Request Sample:

```
<MFEP>
  <MsgHeader>
    <TmStp>2016-08-30T10:19:08</TmStp>
    <TrsInf>
      <SdrCode>1</SdrCode>
      <RcvCode>257</RcvCode>
      <ReqTyp>SYSSVCAVARQ</ReqTyp>
    </TrsInf>
  </MsgHeader>
  <MsgBody>
    <SysAval>
      <Status>UP</Status>
      <Desc>eFAWATEERcom system is available</Desc>
    </SysAval>
  </MsgBody>
  <MsgFooter>
    <Security>xqK9xB87RIMiqODyPswcRN1vazIDUmWplkt/2nyTCIHBqCxaHayL0uKw4QMwngD
      wZKjg1ZdB5ra+KQ+4nRcyWEVlwpZ3cYZp80HNCaVisml1KbwmqJ5x8rok6z9lGu2Ud3bG/uR1B</Security>
  </MsgFooter>
</MFEP>
```

Figure

68 System Service Availability Request Sample

Response Sample:

True/False

6. Business Rules

6.1 Common – Operations

1. Participants are required to send the following parameters:
 - GUID, Customer Code, and Password for 'Authentication' request.
 - GUID, and Token for 'Biller List' request.
 - GUID, Token, and XML file for any other operation request.

C o n 2 U s

Integration Specs

- XML file for any operation response.
2. Elements sent in any request/response should be sorted as specified in each message structure (for details, please refer to the section 'Online Integration Services'), otherwise, message will be rejected.
 3. Any message sent by participant should comply with the XML Schema of the corresponding operation as specified in this document in the section 'ONEIC-TASDEED Integration Services', otherwise, the request/response will be rejected.
 4. If any container exists in any message, then, all its mandatory nodes (child elements) should be sent, otherwise, these child elements should not be sent.
 5. If any optional container is added to any message, then all its mandatory nodes (child elements) should be sent in the message.
 6. When ONEIC-TASDEED sent a response that includes multiple records, probably a 'Result' section will be sent for each record.
 7. Any response for any operation contains an error with severity type = 'Error' in the 'MsgHeader', it means that the transaction/message was not processed successfully.
 8. Any record in any response (in the 'MsgBody') contains an error with severity type 'Error' means that the record was not processed successfully.
 9. Any element with Enumerated Type should have values from the defined lists, otherwise, transaction will be rejected.
 10. Any request/response for any operation should have correct 'Time Stamp' value; the 'TmStp' element is 'date time' format; the date should equal the date of Today and the time should not be in future, otherwise, the message will be rejected.
 11. 'SdrCode', 'RcvCode' elements should be "1" when the sender/receiver in sequence is ONEIC-TASDEED, and it should be equal to the 'participant code' when the sender/receiver is the participant.
 12. In Response Header, 'SdrCode' element should be sent in response if the response is sent by a participant, and 'SdrCode' element will not be in response if the response is sent by ONEIC-TASDEED.
 13. In Response Header, 'RcvCode' element will be sent in response if the response is sent by ONEIC-TASDEED, and 'RcvCode' element should not be in response if the response is sent by a participant.
 14. 'ReqTyp', 'ResTyp' elements in the message header should equal the Operation Shortcut for the corresponding request/response message as defined in section 'ONEIC-TASDEED Operations Types'.

C o n 2 U s

Integration Specs

15. Biller can have only one Category for all his services, and he can choose the appropriate one for its profile through ONEIC-TASDEED Portal, and the category value is not communicated in transaction messages, only the service type is sent.
16. For biller services, 'Thin' or 'Thick' consolidation can be chosen per each service type in the Biller Profile in ONEIC-TASDEED Portal, however, 'Thick' type will not be used in the first phase of ONEIC-TASDEED project.
17. Biller Services will be defined for that Biller in its profile in ONEIC-TASDEED Portal, and message contains 'Service Type' element should use service values as retrieved for that biller in Billers List service, otherwise, message will be rejected.
18. In any message, if the 'BillerCode' element is for 'Inactive Biller', then, the request will be rejected except for 'Remove Customer Billing' request.
19. If the Status of a specific Customer Profile is 'Inactive', then, any request for that profile (Account Upload, Account Inquiry, Add Customer Billing, and Remove Customer Billing) will be rejected.
20. If the Status of a specific Billing Number is 'Inactive', then, any message contains that Billing will be rejected except ('Account Inquiry', 'Remove Customer Billing').

6.2 Header and Footer

1. 'GUID' element should be sent in the 'Header' for any prepaid message request (Prepaid Validation, Biller Prepaid Validation, and Prepaid Payment) either the request is initiated by ONEIC-TASDEED or participants.
2. 'SdrCode' element should be sent in the 'Header' for any 'Request' in any message except 'Biller Prepaid Validation' request message which is initiated by ONEIC-TASDEED.
3. 'SdrCode' element should be sent in the 'Header' for any 'Response' in any message if it is sent by participants, while, if the response message is sent by ONEIC-TASDEED, then, no 'SdrCode' will be sent.
4. 'RcvCode' element will be sent in the 'Header' for any 'Response' in any message if it is sent by ONEIC-TASDEED, while, if the response message is sent by participants, then, 'RcvCode' should not be sent.

C o n 2 U s

Integration Specs

5. In any request header, the 'SdrCode' will equal '1' if the request is sent by ONEIC-TASDEED.
6. In any request header, the 'SdrCode' should equal 'participant code' if the request is sent by participant.
7. In any request header, the 'RcvCode' will equal 'participant code' if the request is sent by ONEIC-TASDEED.
8. In any request header, the 'RcvCode' should equal '1' if the request is sent by participant.
9. In any response header, the 'SdrCode' should equal 'participant code' if the response is sent by participant.
10. In any response header, the 'RcvCode' will equal 'participant code' if the response is sent by ONEIC-TASDEED.
11. In any response header, no 'RcvCode' should be sent if the response is sent by ONEIC-TASDEED.
12. 'Sequence' container is optional to be sent through messages either requests or responses, but it is not used currently at this stage.
13. 'Signature' element in the 'Footer' for any request or response should be sent, except the participant does not have a certificate and CENTRAL BANK approved it.

6.3 Digital Signature

1. Digital signing will be done on a pre-defined 'XPath' (Message Body) and not the full XML file for all operation types, and digital signature is required to be sent in each request/response by all participants (Banks, PSPs, Billers, ONEIC-TASDEED).
2. Participants are required to send ONEIC-TASDEED their certificate that contains the Public Key in order to communicate with the system.
3. Signature for all participants (Banks, PSPs, and Billers) is mandatory, however, Billers that cannot afford digital signing for elements, can proceed without it but after CENTRAL BANK approval, and in this case, those billers should not send the signature element, and they will not receive signature element by ONEIC-TASDEED as well.

6.4 Authentication (Token)

1. Each participant will have a 'Customer Code' and 'Password' for ONEIC-TASDEED, and those credentials will be defined as a part of that participant profile in ONEIC-TASDEED Portal, and can be used for authorizing that participant to be able to connect with ONEIC-TASDEED.
2. Once the participant gains the Token, it should be part of any future request, and if the participant calls the 'Authenticate' web service again, the Token will be renewed with a new expiration date.
3. A valid and not-expired Token should be sent with each XML request sent by participants.
4. If a participant asked to change its password, and ONEIC-TASDEED changed/reset the password, then, its token will be auto-killed, and requests will be rejected "InvalidToken", and participant should request a new one using Authentication to communicate again successfully with ONEIC-TASDEED. Once a participant Token is killed, requests will be rejected "InvalidToken", and participant should request a new one using Authentication to communicate again successfully with ONEIC-TASDEED.
5. Once any participant is deactivated (Inactive Status), its Token will be auto-killed, thus, Inactive participant is not allowed to send requests for any operation, requests will be rejected 'Invalid Token'.
6. If ONEIC-TASDEED activated/reactivated any participant, the participant should call the 'Authenticate' web service to accept its requests.
7. Deleted participants are not allowed to send requests for any operation, Token will be auto-killed and requests will be rejected.
8. For any request initiated by ONEIC-TASDEED, no Token will be sent.

6.5 Billers List – V.3

1. Billers List response will retrieve the updated list of all 'Active' Billers, while the 'Inactive' or 'Deleted' Billers will not be retrieved.

C o n 2 U s

Integration Specs

2. Billers list response will retrieve records sorted per biller level, service level, and prepaid category level, as the following:
 - a. Billers sorting: billers will be sorted based on the number of succeeded payment each biller get (accumulated for all payments received by the system for that biller) (most to least)
 - b. Services sorting: services will be sorted based on the number of succeeded payment each service get per biller (accumulated for all payments received by the system for that service under that biller) (most to least)
 - c. Prepaid Category sorting: prepaid categories will be sorted based on the prepaid category itself from the smallest value to the largest value (e.g.: JD_1, JD_3, JD_5, etc...)
 - d. The billers List records sorting will be updated periodically so the customers will not find new sorting changes every day.
 - e. It is highly recommended for Banks/PSPs to update and sort their lists (billers, services, prepaid categories) on their payment channels based on the sorting retrieved in billers list response.
3. If the 'IntegrationType' of a biller is 'Offline', then, it will be retrieved in biller list response, but Banks/PSPs should not send any request targeting that biller, otherwise request will be rejected.
4. If 'Payment Type' = Postpaid, then, the Banks/PSPs should invoke Bill Inquiry service and Bill Payment Service.
5. If 'Payment Type' = Prepaid, then, the Banks/PSPs should invoke Prepaid Validation Service and Prepaid Payment Service.
6. If 'Payment Type' = Postpaid, then, ONEIC-TASDEED will not send 'BillingNoRequired' and 'ContainsPrepaidCats' elements for that service.
7. If 'Payment Type' = Prepaid, then, 'BillingNoRequired' and 'ContainsPrepaidCats' elements will be sent for that service with values true or false for each.
8. If 'Payment Type' = Prepaid and 'ContainsPrepaidCats' = true, then, ONEIC-TASDEED will send 'PrepaidCategories' container with its elements for that service.
9. If 'Payment Type' = Prepaid and 'BillingNoRequired'= true, then, ONEIC-TASDEED will send 'BillingInfo' container with its elements for that service.
10. If 'Payment Type' = Prepaid and 'BillingNoRequired'= false, then, ONEIC-TASDEED will not send 'BillingInfo' container with its elements for that service. It is recommended for Banks/PSPs to use the short names/descriptions provided in biller list response V.3



C o n 2 U s

Integration Specs

(EnShortName, ArShortName, EnShortDesc, ArShortDesc, etc...) to be displayed in channels that have size limitations such as ATM.



6.6 Account Upload Process (Customer Profile)

1. The three elements (ID Type, ID, and Nationality) combines the unique ID of any Customer Profile. Any Account Upload request should contains these three elements.
2. If the account upload request contains a combination of (ID Type, ID, and Nationality) that already exist, then, ONEIC-TASDEED will overwrite the exist record and update the optional values (Name, Phone, Address, Email) and if any of the optional values not sent in the request, then, the old values will be kept for those elements.
3. Customer profiles will be built up by inserting the basic info sent in 'Account Upload' request, and will add billing numbers under that customer profile by 'Add Customer Billing' request, and the bills sent for that billing number will be displayed under it as well with its corresponding payments.
4. If the account upload request succeeded (Error Code = 0) and a new profile is created, then, ONEIC-TASDEED will send a new 'JOEBPPSNo' value in the response.
5. On creating a Customer Profile, by sending 'Account Upload' request, the profile/account status will be 'Active' by default.
6. Changing the Customer Profile Status (Active/Inactive) will be done manually and only by using ONEIC-TASDEED Portal by the System Operations team.
7. In Account Upload process, there is no business validation between Official ID Type and Nationalities values.
8. Currently, ONEIC-TASDEED does not allow bulks for Account Upload service.

6.7 Account Inquiry

1. The three elements (ID Type, ID, and Nationality) under 'CustId' container combines the unique ID of any Customer Profile. Any Account Inquiry request should contain either 'CustId' container or 'JOEBPPSNo' element. If the request contains none of them or both of them, then, it will be rejected.
2. If the request has Error Code = '0', then, ONEIC-TASDEED will send 'CustId', 'JOEBPPSNo', and 'RecCount' elements in the response.
3. If the request has Error Code = '0' and 'RecCount' > '0', then, ONEIC-TASDEED will send 'BillingsRec' element in the response.



C o n 2 U s

Integration Specs

4. Account Inquiry response will retrieve all Billings that were added to that customer profile regardless of the billings status (Active and Inactive Billings).
5. Account Inquiry response will retrieve billings regardless of the Biller Status (Active, Inactive Biller).
6. Currently, Account Inquiry response will retrieve billings of Postpaid services only (Retrieving billings of prepaid services is to be implemented in future versions).
7. Currently, ONEIC-TASDEED does not allow bulks for Account Inquiry service.

6.8 Add Customer Billing

1. Add Customer Billing request should contain 'JOEBPPSNo' element, otherwise, it will be rejected.
2. The 'JOEBPPSNo' value sent in the request should be exist at eFAWTAEERcom, otherwise, request will be rejected.
3. If the request has Error Code = '0', then, ONEIC-TASDEED will send 'BillingsRec' element in the response.
4. Multiple customer profiles can contain the same billing number.
5. The status of the billing account will be the same under all customer profiles that have it.
6. If Add Customer Billing request contains a billing that already exists under the specified customer profile, then, a message will be displayed 'This Billing Already Exist Under This Profile'.
7. Currently, Add Customer Billing services can be used for Postpaid services only (Add billings for prepaid services is to be implemented in future versions).
8. Each 'Add Customer Billing' request should be sent for only one 'customer profile'/'account'.
9. Currently, ONEIC-TASDEED does not allow bulks for Add Customer Billing service.

6.9 Remove Customer Billing





C o n 2 U s

Integration Specs

1. 'Remove Customer Billing' request should contain JOEBPPS No., otherwise, it will be rejected.
2. If the first 'Result' container in the 'MsgBody' has Error Code = '0', then, ONEIC-TASDEED will send 'BillingsRec' element in the response, otherwise, 'BillingsRec' will not be retrieved.
3. The 'JOEBPPSNo' value sent in the request should be exist at eFAWTAEERcom, otherwise, request will be rejected.
4. If the Billing status is 'Inactive', then, 'Remove Customer Billing' request will be accepted.
5. In Remove Customer Billing request, if the biller code is for Inactive biller, then, the request will be accepted.
6. If 'Remove Customer Billing' request contains a billing number that does not exist under the specified customer profile, then, request will be rejected.
7. Removing billing number from a specific customer profile will not affect any other customer profile(s).
8. Currently, Remove Customer Billing services can be used for Postpaid services only (Remove billings for prepaid services is to be implemented in future versions).
9. Each 'Remove Customer Billing' request should be sent for only one customer profile.
10. Currently, ONEIC-TASDEED does not allow bulks for Remove Customer Billing service.

6.10 Online Bill Upload and Offline Bill Batch

1. The four elements (Bill No, Billing No, Biller Code, Service Type) combines the unique identification of any Bill, and any bill upload or bill batch request contains existing values for these four elements will result in updating the bill, otherwise, a new bill record will be added.
2. Bill upload or bill batch request should have 'BillStatus'= 'BillNew' or 'BillUpdated', otherwise, it will be rejected.
3. Bill upload or bill batch request can be sent on any bill exists in ONEIC-TASDEED (sent previously) regardless of its status in ONEIC-TASDEED and it is supposed to have the status 'BillUpdated' in the request, and if it is sent for the first time, it is supposed to have the status 'BillNew'.



C o n 2 U s

Integration Specs

4. Any bill uploaded will be reflected on all Customer Profiles that have that billing (billing number, biller code, service type), and any payment occurred on that bill, will be reflected on the bill wherever it exists.
5. There is no business validation on 'Bill Type' values (Recurring, OneOff).
6. 'Service Type' sent in request should have 'Payment Type' = Postpaid, otherwise, request will be rejected.
7. Biller can send multiple bill records (Bills) in the same bill upload or bill batch request.
8. The bill dates sent in bill upload or bill batch request have the following rules which should be matched, otherwise, the request will be rejected:
 - 'Issue Date' should be less than or equal 'Open Date'
 - 'Open Date' should be less than or equal 'Due Date'
 - 'Due Date' should be less than 'Expiry Date'
 - 'Expiry Date' should be less than or equal 'Close Date'
9. The bill dates have restrictions as follows:
 - 'Issue Date' should not be in future
 - (Issue Date, Open Date, or Due Date) should not have a value older than '15' years
 - 'Expiry Date' should not be in past
 - 'Close Date' should not be in past
10. In case of 'Thin' model, the bill number should be equal to the billing number, meaning that, the one billing number cannot have more than one bill number.
11. In case of 'Thin' model, any bill upload or bill batch process for an existing bill will result in replacing that bill, not adding the due amount in the new request to the existing amount, biller himself should do the aggregation and send the new amount, ONEIC-TASDEED is not responsible for aggregating the amounts.
12. In PmtConst, 'Upper' and 'Lower' values represents amounts not percentage values, e.g. '80.555'.
13. In PmtConst, 'Upper' and 'Lower' values should be greater than 0. The 'Lower' and 'Upper' values in 'PmtConst' will be validated in case the 'Due Amount' > 0 regardless to the value of 'AllowPart'.
14. If PmtConst element is not sent by the biller (does not exist in the message), then, ONEIC-TASDEED will consider the bill to be fully paid (Exact) (partial payments or advance payments are not allowed for this bill).
15. If 'AllowPart' = false, and the 'Due Amount' = 0 or less, then the 'Upper' and 'Lower' values will not be validated.



C o n 2 U s

Integration Specs

16. If 'AllowPart'= false, then, the (Due Amount, Upper, Lower) values of the bill should equal each other, and ONEIC-TASDEED will consider the bill to be fully paid (Exact) in case of the 'Due Amount' is larger than 0. The Due Amount of the bill should be within the range of 'Max' and 'Min' values defined for that Service Type of that biller in his Profile in ONEIC-TASDEED Portal, and in case of 'Min' = '0', then, Due Amounts ='0' or less are allowed.
17. If 'AllowPart'= false or if no 'PmtConst' container sent for a bill, then, the 'Due Amount' sent for that bill in bill upload request can be with any amount ('0' or more or less).
18. If the bill due amount= '0' or less and biller does not want to allow advance payments, then, biller should not sent 'PmtConst' container or he can send it with 'AllowPart' = 'false' for that bill.
19. Any bill has Due Amount='0' or less, it will be retrieved in bill inquiry response as Due Amount='0' if 'Fees on Biller'= true and 'Allow Part'=true.
20. Any bill has Due Amount='0' or less, it will be retrieved in bill inquiry response as Due Amount='FeesAmt' if 'Fees on Biller' = false and 'Allow Part'=true.
21. Any Biller can send Bill Upload or Bill Batch request with 'Sub-Payments' for any bill for any service.
22. ONEIC-TASDEED will define a new flag for billers to mention if it Allow advance payment or not.
23. ONEIC-TASDEED will define a new flag for billers to mention if it 'Allow multiple settlement banks' or not:
 - If the biller has 'Allow multiple settlement banks'= false, then, the 'SetBnkCode' in each sub-payment should equal that biller settlement bank code defined in ONEIC-TASDEED.
 - If the biller has 'Allow multiple settlement banks'= true, then, biller can send 'SetBnkCode' different than its settlement bank code defined in ONEIC-TASDEED for bills sub-payments, but all sub-payments in the same bill should have the same 'SetBnkCode' with either same or different Accounts.
 - If the biller has 'Allow multiple settlement banks'= false, then, biller can send bills with sub-payment(s) or without sub-payment(s).
 - The biller can send bill with only one sub-payment in case of 'AllowPart' = 'true'.



C on 2 Us

Integration Specs

- If the biller has 'Allow multiple settlement banks'= true, then, biller should send sub-payment(s) ('SubPmts' container) for any of his bills.
 - If the biller has 'Allow multiple settlement banks'= true, then, biller should not send 'AllowPart'= true for any of his bills.
24. The total 'Amount' values in sub payments should equal the 'Due Amount' of that bill.
25. The account number 'AcctNo' in each sub-payment should have the IBAN Number including the bank BIC code in its (5-9) digits of the value (e.g. 'JOxxCENTRAL BANKOxxxxxxxxxxxxxxxxxxxx') and it should equal the BIC code of the 'SetBnkCode' sent, otherwise, bill will be rejected.

6.11 Bill Pull

1. The four elements (Bill No, Billing No, Biller Code, Service Type) combines the unique identification of any Bill, and any Bill Pull response contains existing values for these four elements will result in updating the bill, otherwise, a new bill record will be added.
2. In bill pull response, biller should send 'BillsRec' element if 'RecCount' > '0' and fill all the required elements and containers such as : 'Result', 'AcctInfo', 'DueAmount', 'PmtConst', etc ...
3. In Bill Pull response, 'BillStatus' should be 'Bill New' or 'BillUpdated', otherwise, response will not be processed.
4. Any bill pulled will be reflected on all Customer Profiles that have that billing (billing number, biller code, service type), and any payment occurred on that bill, will be reflected on the bill wherever it exists.
5. 'Billing No' and 'Service Type' sent in response should match the values sent in request, otherwise, response will not be processed.
6. In case of 'Thin' model, the bill number should equal the billing number, meaning that, the one billing number cannot have more than one bill number.
7. In case of 'Thin' model, any bill pull process for an existing bill will result in replacing that bill, not adding the due amount in the new request to the existing amount, biller himself should do the aggregation and send the new amount, ONEIC-TASDEED is not responsible for aggregating the amounts.
8. There is no business validation on 'Bill Type' values (Recurring, OneOff).

C o n 2 U s

Integration Specs

9. The bill dates sent in bill pull response have the following rules which should be matched, otherwise, response will not be processed:
 - 'Issue Date' less than or equal 'Open Date'
 - 'Open Date' less than or equal 'Due Date'
 - 'Due Date' less than 'Expiry Date'
 - 'Expiry Date' less than or equal 'Close Date'
10. The bill dates should match the following restrictions, otherwise, response will not be processed:
 - 'Issue Date' should not be in future
 - ('Issue Date, Open Date, or Due Date) should not have a value older than '15' years
 - 'Expiry Date' should not be in past
 - 'Close Date' should not be in past
11. In case of 'Thin' model, the records count in response should be either '0' or '1'.
12. The Due Amount of the bill should be within the range of 'Max' and 'Min' values defined for that Service Type of that biller in his Profile in ONEIC-TASDEED Portal, and in case of 'Min' = '0', then, Due Amounts= '0' or less are allowed.
13. In PmtConst, 'Upper' and 'Lower' values represents amounts not percentage values, e.g. '80.555'.
14. In PmtConst, 'Upper' and 'Lower' values should be greater than 0.
15. The 'Lower' and 'Upper' values in 'PmtConst' will be validated in case the 'Due Amount' > 0 regardless to the value of 'AllowPart'.
16. If PmtConst element is not sent by the biller (does not exist in the message), then, ONEIC-TASDEED will consider the bill to be fully paid (Exact) (no partial payments or advance payments are allowed for this bill).
17. If 'AllowPart'= false, then, the (Due Amount, Upper, Lower) values of the bill should equal each other, and ONEIC-TASDEED will consider the bill to be fully paid (Exact) (no partial payments or advance payments are allowed for this bill).
18. If 'AllowPart'= false or if no 'PmtConst' container sent for a bill, then, the 'Due Amount' sent for that bill in bill pull response can be with any amount ('0' or more or less).
19. If 'AllowPart'= true, then, Bills with 'Lower' value ='0' or less are not allowed, 'Lower' should be larger than '0'.
20. If 'AllowPart'= true, then, Bills with 'Upper' value ='0' or less are not allowed, 'Upper' should be larger than '0'.

C o n 2 U s

Integration Specs

21. If 'AllowPart'= false, and Due Amount ='0' or less, then, Lower and Upper values can be sent as '0' or less (equal the due amount), but no payments will be allowed for such bills.
22. If the biller wants to permit his customers to pay in advance but does not want his customers to pay partially, then, he should send 'Lower'= 'Due Amount', so, no partial payments can be done on that bill while advance payment can be done.
23. If the biller wants to allow partial payments and not allow advance payments, then, he should send 'Upper'= 'Due Amount', so, no advance payments can be done on that bill while partial payment can be done.
24. If the bill due amount= '0' or less and biller does not want to allow advance payments, then, biller should not sent 'PmtConst' container or he can send it with 'AllowPart' = 'false' for that bill.
25. Any bill has Due Amount='0' or less, it will be retrieved in bill inquiry response as Due Amount='0' if 'Fees on Biller'= true.
26. Any bill has Due Amount='0' or less, it will be retrieved in bill inquiry response as Due Amount=' FeesAmt ' if 'Fees on Biller'= false and 'Allow Part'=true.
27. Any Biller can send Bill Pull response with 'Sub-Payments' for any bill for any service.
28. ONEIC-TASDEED will define a new flag for billers to mention if it Allow advance payment or not.
29. ONEIC-TASDEED will define a new flag for billers to mention if it 'Allow multiple settlement banks' or not:
- If the biller has 'Allow multiple settlement banks'= false, then, the 'SetBnkCode' in each sub-payment should equal that biller settlement bank code defined in ONEIC-TASDEED.
 - If the biller has 'Allow multiple settlement banks'= true, then, biller can send 'SetBnkCode' different than its settlement bank code defined in ONEIC-TASDEED for bills sub-payments, but all sub-payments in the same bill should have the same 'SetBnkCode' with either same or different Accounts.
 - If the biller has 'Allow multiple settlement banks'= false, then, biller can send bills with sub-payment(s) or without sub-payment(s).
 - The biller can send bill with only one sub-payment in case of 'AllowPart' = 'true'.

C o n 2 U s

Integration Specs

- If the biller has 'Allow multiple settlement banks'= true, then, biller should send sub-payment(s) ('SubPmts' container) for any of his bills.
- If the biller has 'Allow multiple settlement banks'= true, then, biller should not send 'AllowPart'= true for any of his bills.

30. The total 'Amount' values in sub payments should equal the 'Due Amount' of that bill.

31. The account number 'AcctNo' in each sub-payment should have the IBAN Number including the bank BIC code in its (5-9) digits of the value (e.g. 'JOxxCENTRAL BANKOxxxxxxxxxxxxxxx') and it should equal the BIC code of the 'SetBnkCode' sent, otherwise, bill will not be processed.

6.12 Bill Inquiry

1. ONEIC-TASDEED does not allow bulks for bill inquiry request.
2. Currently, Bill inquiry request should include the biller code, service type, and billing number.
3. It is strongly recommended to have a double entry on the Billing Number field in the interfaces of the Banks and PSPs channels, thus minimal entry error can occur by the customer.
4. In bill inquiry response, ONEIC-TASDEED will send 'BillsRec' element if 'RecCount' > '0'.
5. In bill inquiry response, ONEIC-TASDEED will send 'BillStatus' = BillNew, BillUpdated, ,BillPartiallyPaid or BillOverPaid
6. If 'IncPayments' is sent in request and it has 'true' value, then, ONEIC-TASDEED will send 'BillPmts' element in its response if the related bill already has payments.
7. In case of 'Thin' model, Bill Inquiry might retrieve no records or maximum one record for that billing.
8. If the 'Due Amount' of the bill is '0' or less, and 'AllowPart'= false, then, bill record will be retrieved in bill inquiry response with 'Due Amount'= '0' but no payments will be allowed for it.
9. If the 'Due Amount' of the bill is '0' or less, and 'AllowPart'=true, then, bill record will be retrieved in bill inquiry response with 'Due Amount'= '0' in case of Fees on Biller, and with 'Due Amount'= 'FeesAmt' in case of Fees on Customer.
10. Bill inquiry request can have 'Date Range' elements to inquire about a bill in a specific period in regards of its 'Issue Date'.
11. If 'Fees on Biller' = true, then, the fees amount retrieved in response will be '0'.
12. If 'Fees on Biller'= false and 'Due Amount' is larger than '0', then, the Fees Amount retrieved in response will be larger than '0' (most probably), and the 'Due Amount' will include the fees amount as well.
13. If 'Fees on Biller'= false and 'Due Amount'= '0' or less, and 'Allow Part'=true, then, the Fees Amount will be larger than '0' (most probably), and the 'Due Amount' will include the fees amount as well.
14. If 'Fees on Biller'= false and 'Due Amount'= '0' or less, and 'Allow Part'=false, then, the Fees Amount will be '0'.

C o n 2 U s

Integration Specs

15. For any reason, if an error occurred from the Biller side in Bill Pull response, then, ONEIC-TASDEED will not send any bill record(s) in the corresponding Bill Inquiry response.
16. Bills that have 'Expiry Date' in past will be retrieved in bill inquiry response, while the bills that have 'Close Date' in past will not be retrieved in bill inquiry response.
17. ONEIC-TASDEED added a new flag for each Bank/PSP 'IsBillInquiryRefNoRequired', if the Bank/PSP has 'IsBillInquiryRefNoRequired'= true, then, ONEIC-TASDEED will send 'InqRefNo' element with unique value in each bill inquiry response for that Bank/PSP, otherwise, 'InqRefNo' element will not be sent.
18. If 'PmtConst' elements are not retrieved in the bill inquiry response, then, the bill is supposed to be fully paid (Exact).
19. In bill inquiry response, if 'AllowPart' is 'false', then, the bill is supposed to be fully paid (Exact) (partial payments or advance payments are not allowed).
20. In bill inquiry response, if 'AllowPart' is 'true' and 'Lower' value is less than the 'Due Amount' value, then, the bill can be 'Partially Paid'.
21. In bill inquiry response, if 'AllowPart' is 'true' and 'Upper' value is larger than the 'Due Amount' value, then, the bill can be 'Over Paid'.
22. Any bill has Due Amount='0' or less, it will be retrieved in bill inquiry response as Due Amount='FeesAmt' if 'Fees on Biller'= false and 'Allow Part'=true.
23. 'Include Payments'= true can be sent for 'Thin' service types only, otherwise, no payment records will be retrieved for that bill.
24. If 'Include Payments' is sent for a 'Thin' service type, it will retrieve all payments done on that bill (up to maximum twelve '12' records only – the latest payment records for that billing).
25. 'Include Paid Bills'= true can be sent for 'Thick' service types only (which is not used currently at this phase), if 'Include Paid Bills' is sent for a 'Thin' service type, then, no impact will occur on the response (no payment records will be retrieved).
26. Any new inquiry will make a specific billing number exceeds any defined caps limit(s), then, the inquiry will return a warning message by ONEIC-TASDEED gateway.
27. Banks and PSPs must send the 'JOEBPPSNo' / 'IdType', 'Id', 'Nation' element into 'Payer Info' details in any bill inquiry request, so, the inquiries can be connected to customer profiles. (The JOEBPPSNo / 'IdType', 'Id', 'Nation' element will be optional for specific period and then will be required for each transaction)The integration with the CSPD (Civil Status and Passport Department of Jordan) requires that the Banks/PSPs to send

C on 2 Us

Integration Specs

the details of the customers in the required container <PayerInfo>. Banks/PSPs should take into consideration the following:

- a. Banks should clean all the current customer data and remove any reference to wrong data for (ID, IDType, Nationality) **before November 1, 2016** and this process should be arranged and confirmed with MadfusatCom Team.
- b. Sending this data (ID, IDType, and Nationality) will be optional ~~till November 1, 2016~~. After this date; all inquiry and payment transactions that do not contain this data will be rejected.

6.13 Bill Payment

1. Banks/PSPs should send bill inquiry request before any bill payment request.
2. Bill Payment request should have a bill already exists in ONEIC-TASDEED.
3. Bank/PSP can send Bill Payment requests with 'Payment Status' equal 'PmtNew' only, otherwise, payment will be rejected.
4. It is strongly recommended to have a double entry on the 'Billing Number' field in the interface of the Bank/PSP channel, thus minimal entry error can occur by the customer.
5. If 'PaymentType' element is sent in bill payment request, then, it should have the value 'Postpaid', otherwise, request will be rejected.
6. No validation will be done on 'Currency' values currently.
7. The 'Service Type' value of the payment request should match the Service Type value sent for the corresponding bill.
8. If a bill has 'Open Date' in future, then, any payment request for that bill will be rejected, if its 'Open date' is in past, then, payment request will be accepted regardless of its 'Due Date'.
9. If a bill does not have 'Open Date' defined but has 'DueDate' and it is in future, then, any payment request for that bill will be rejected, but if the Due Date is in past, payment request will be accepted.
10. If Allow Partial is 'false', then, the value of 'Paid Amount' should equal the 'Due Amount' sent in bill inquiry response in case of the 'Due Amount' is larger than 0.
11. If Allow Partial is 'true' and the value of 'Due Amount' equal 'Fees Amount', then the 'Paid Amount' should be greater than 'Due Amount'.

C o n 2 U s

Integration Specs

12. If Allow Partial is 'true', then, the value of 'Paid Amount' should be within the range of 'Upper' and 'Lower' values sent in bill inquiry response.
13. It is highly recommended to send the 'JOEBPPSNo' element in and 'Payer Info' details in any bill payment request, so, payments can be connected to customer profiles.
14. The 'JOEBPPSNo' value to be sent in 'Payer Info' should hold the same 'JOEBPPSNo' provided to the Bank/PSP in the Account Upload response and/or Account Inquiry response.
15. Bank/PSP can send multiple bill payment records in the same request.
16. ONEIC-TASDEED added a new flag for each Bank/PSP 'IsBillInquiryRefNoRequired', if the Bank/PSP has 'IsBillInquiryRefNoRequired'= true, then, Bank/PSP should send 'InqRefNo' element in each bill payment request, otherwise, request will be rejected.
17. The 'InqRefNo' element sent in bill payment request should have the same value received in the latest bill inquiry transaction done for that bill, otherwise, request will be rejected.
18. If the Bank/PSP has 'IsBillInquiryRefNoRequired'= false, then, Bank/PSP should not send 'InqRefNo' element in bill payment request, otherwise, request will be rejected.
19. If Allow Partial is 'true' and 'Upper' value is larger than 'Due Amount', then, the user can pay more than the due amount, and the bill status will be 'Over Paid'.
20. If Allow Partial is 'true' and 'Lower' value is less than 'Due Amount', then, the user can pay less than the due amount, and the bill status will be 'Bill Partially Paid'.
21. If the bill due amount is '0' or less and AllowPart= 'false' for that bill, then, any bill payment request will be rejected.
22. Any bill payment request will be rejected in case of the 'Paid Amount' ='0' or less or 'Paid Amount' <= 'Fees Amount'.
23. The Fees Amount charged on each succeeded payment represents the total fees deducted for that payment transaction, not for that bill, either the fees is on biller or on customer.
24. If the 'Expiry Date' or 'Close Date' of the bill is in past, then, any bill payment request will be rejected.
25. In Bill Payment request, the 'BankTrxID' sent should be unique per payment, otherwise, request will be rejected.
26. It is recommended that Banks/PSPs display the 'BankTrxID' for the customer after each bill payment process.

Con 2 Us

Integration Specs

27. For each successful Bill Payment request, ONEIC-TASDEED will send a new 'JOEBPPSTrx' in the response.
28. 'Process Date' sent by Bank/PSP in Bill Payment request should have the current date (Today), and it should have time equal now or less, otherwise, the request will be rejected.
29. For each successful Bill Payment request, ONEIC-TASDEED will send a 'STMT Date' for that payment in the response.
30. 'STMT Date' sent by ONEIC-TASDEED in Bill Payment response will have either the current date (T) or the date of the next business day (T+1).
31. Any new payment will make a specific billing number exceeds any defined caps limit(s), then, the payment will be rejected by ONEIC-TASDEED gateway.
32. Banks and PSPs must send the 'JOEBPPSNo' / 'IdType', 'Id', 'Nation' element into 'Payer Info' details in any bill payment request, so, the payments can be connected to customer profiles. (The JOEBPPSNo will be optional for specific period and then will be required for each transaction) The integration with the CSPD (Civil Status and Passport Department of Jordan) requires that the Banks/PSPs to send the details of the customers in the required container <PayerInfo>. ~~Banks/PSPs should take into consideration the following:~~
 - a. ~~Banks should clean all the current customer data and remove any reference to wrong data for (ID, IDType, Nationality) before November 1, 2016 and this process should be arranged and confirmed with MadfootatCom Team.~~
 - b. ~~Sending this data (ID, IDType, and Nationality) will be optional till November 1, 2016. After this date, all inquiry and payment transactions that do not contain this data will be rejected.~~

6.14 Prepaid Validation

1. For the selected service, if 'BillingNoRequired' = true, then, 'BillingNo' should be sent in the request, and if false, then, it should not be sent in the request.
2. ONEIC-TASDEED will send 'BillingNo' element in response if error code = '0' and if 'BillingNoRequired' = true.
3. For the selected service, if 'ContainsPrepaidCats' = true, then, 'PrepaidCat' should be sent in the request, and if false, then, it should not be sent in the request.

C o n 2 U s

Integration Specs

4. For any Service Type, if the element 'ContainsPrepaidCats' = true, then, the Due Amount should not be sent in the request.
5. For any Service Type, if the element 'ContainsPrepaidCats' = false, then, the Due Amount should be sent in the request.
6. If Error Code = '0' and 'FeesOnBiller'=false, then, ONEIC-TASDEED will send 'FeesAmt' element in the response, and the 'DueAmt' value will include the 'FeesAmt'. And in this case, Banks/PSPs should display the Due Amount as is (including the fees amount); fees amount should not be declared as a separate value.
7. If Error Code = '0', eFAWTAEERcom will send 'Validation Code' in response to be used in the Prepaid Payment request.
8. If Error Code = '0', eFAWTAEERcom will send 'AcctInfo', 'DueAmt', 'ServiceTypeDetails' elements in response.
9. Request for any service with 'PaymentType'= Postpaid will be rejected.
10. No bulks allowed for this service (only one record).
11. For any reason, if an error occurred from biller side in Biller Prepaid Validation transaction, an error will be sent in the corresponding Bank/PSP Prepaid Validation response.
12. Banks and PSPs must send the 'JOEBPPSNo'/ 'IdType', 'Id', 'Nation' element into 'Payer Info' details in any prepaid validation request, so, the requests can be connected to customer profiles.
13. The 'JOEBPPSNo' value to be sent in 'Payer Info' should hold the same 'JOEBPPSNo' provided to the Bank/PSP in the Account Upload response and/or Account Inquiry response.
14. The integration with the CSPD (Civil Status and Passport Department of Jordan) requires that the Banks/PSPs to send the details of the customers in the required container <PayerInfo>. ~~Banks/PSPs should take into consideration the following:~~
 - a. ~~Banks should clean all the current customer data and remove any reference to wrong data for (ID, IDType, Nationality) before November 1, 2016 and this process should be arranged and confirmed with MadfooratCom Team.~~
 - b. ~~Sending this data (ID, IDType, and Nationality) will be optional till November 1, 2016. After this date, all inquiry and payment transactions that do not contain this data will be rejected.~~

6.15 Biller Prepaid Validation

1. For any service, if Biller Prepaid Validation is 'Required' for that service and it has categories (Contains Prepaid Cats=true), then, no values/amounts for its categories will be pre-defined at ONEIC-TASDEED. Value/Amount for that category will be retrieved from biller per each request.
2. ONEIC-TASDEED will send 'BillingNo' element in request if 'BillingNoRequired' = true.
3. If Error Code = '0' and 'BillingNoRequired' = true, then, the Biller should send 'BillingNo' in response and its value should match the one sent in request.
4. If Error Code = '0', then, the Biller should send 'DueAmt', 'ValidationCode', 'ServiceTypeDetails' elements in the response, otherwise, these elements should not be sent.
5. If Error Code = '0' and 'ContainsPrepaidCats' = true for the service, then, the Biller should send 'PrepaidCat' element in the response, otherwise, it should not be sent.
6. For any service, if Biller Prepaid Validation is 'Not Required' for that service and it has categories (Contains Prepaid Cats=true), then, values/amounts for its categories will be pre-defined at ONEIC-TASDEED.
7. For any service, if 'ContainsPrepaidCats' = 'false', then, ONEIC-TASDEED will send 'Due Amount' element in the request, and if it is 'true', then, 'Due Amount' element will not be sent in request.
8. If 'Due Amount' is sent in request, then, the response should include the same 'Due Amount' in case of error code = '0'.
9. If the 'Due Amount' is not sent in request, then, in response and if error code = '0', the biller should send correct 'Due Amount' and it should be related to the category sent in request.
10. For any service, if 'ContainsPrepaidCats' = true, then, ONEIC-TASDEED will send 'PrepaidCat' element in the request, otherwise, it will not be sent.
11. No bulks allowed for this service (only one record).
12. For any service, if Biller Prepaid Validation is Required for that service, then, ONEIC-TASDEED will send a biller prepaid validation request after each Bank/PSP prepaid validation request received from Bank/PSP side, otherwise, ONEIC-TASDEED will validate the Bank/PSP request without referring to the biller.

13. For any reason, if an error occurred from biller side in Biller Prepaid Validation response, an error will be sent in the corresponding Bank/PSP Prepaid Validation response.

6.16 Prepaid Payment

1. For any service, if 'ContainsPrepaidCats' = true, then, Prepaid Payment request should include 'PrepaidCat' element, and if 'false', then, request should not contain it.
2. For any service, if 'BillingNoRequired' = true, then, 'BillingNo' should be sent in the request, and if 'false', then, request should not contain it.
3. For any service, if 'BillingNoRequired' = true, ONEIC-TASDEED will send ' BillingNo' element in the response.
4. 'PmtStatus' element should have the value 'PmtNew', otherwise, request will be rejected.
5. No validation will be done on 'Currency' values between the payment currency and bank defined currency.
6. The values of (biller code, service type, billing number 'if exist', prepaid category 'if exist', due amount, validation code) sent in prepaid payment request should equal the values sent by the same Bank/PSP in the corresponding prepaid validation message, otherwise, Bank/PSP should send prepaid validation request first.
7. Prepaid payment request will be rejected if the related prepaid validation response was not succeeded.
8. Any validation code can have one successful payment transaction, any other payment request will be rejected.
9. Any prepaid payment request will be rejected in case of the 'Paid Amount' ='0' or less or 'Paid Amount' <= 'Fees Amount'.
10. The 'ProcessDate' sent in Prepaid Payment request should have the current date (T), otherwise, the request will be rejected.
11. For each successful prepaid payment request, ONEIC-TASDEED will send a 'STMT Date' element for that payment in the response.
12. 'STMTDate' sent by ONEIC-TASDEED in Prepaid Payment response will have either the current date (T) or the date of the next business day (T+1).

C o n 2 U s

Integration Specs

13. In Prepaid Payment request, the 'BankTrxID' sent should be unique per payment, otherwise, request will be rejected.
14. It is recommended that Banks/PSPs display the 'BankTrxID' for the customer after each prepaid payment process.
15. For each successful Prepaid Payment request, ONEIC-TASDEED will send a new 'JOEBPPSTrx' in the response.
16. Request for any service with 'PaymentType'= Postpaid will be rejected.
17. No bulks allowed for this service (only one payment record).
18. The 'Paid Amount' sent in prepaid payment request should equal the 'Due Amount' sent in the same request.
19. If 'PaymentType' element is sent in prepaid payment request, then, it should have the value 'Prepaid', otherwise, request will be rejected.
20. Banks and PSPs must send the 'JOEBPPSNo' / 'IdType', 'Id', 'Nation' element into 'Payer Info' details in any prepaid payment request, so, the payments can be connected to customer profiles. (The JOEBPPSNo will be optional for specific period and then will be required for each transaction).
21. The integration with the CSPD (Civil Status and Passport Department of Jordan) requires that the Banks/PSPs to send the details of the customers in the required container <PayerInfo>. ~~Banks/PSPs should take into consideration the following:~~
 1. ~~Banks should clean all the current customer data and remove any reference to wrong data for (ID, IDType, Nationality) before November 1, 2016 and this process should be arranged and confirmed with MadfooratCom Team.~~
 2. ~~Sending this data (ID, IDType, and Nationality) will be optional till November 1, 2016. After this date; all inquiry and payment transactions that do not contain this data will be rejected.~~

6.17 Capacity Constraints

1. For each Bank, Two Capacity constraints will be defined (values will be defined by CENTRAL BANK) as the following:
 - Maximum Payment Amounts: represents the maximum allowed amount for succeeded collected payments (Postpaid and Prepaid) through ONEIC-TASDEED as a 'Paying Bank'.

C o n 2 U s

Integration Specs

- Maximum Number of Transactions: represents the maximum allowed number of succeeded payments transactions (Postpaid and Prepaid) through ONEIC-TASDEED as a 'Paying Bank'.
- 2. For each PSP, Two Capacity constraints will be defined (values will be defined by CENTRAL BANK) as the following:
 - Maximum Payment Amounts: represents the maximum allowed amount for succeeded collected payments (Postpaid and Prepaid) through ONEIC-TASDEED.
 - Maximum Number of Transactions: represents the maximum allowed number of succeeded payments transactions (Postpaid and Prepaid) through ONEIC-TASDEED.

6.18 Payment Inquiry

1. Payment Inquiry request should be sent by the initiator of the payment itself, no participant can inquire about a payment done by another participant.
2. 'BillNo' element should be sent in payment inquiry request if 'PaymentType' = Postpaid, otherwise, it should not be sent.
3. 'BillNo' element will be sent in response if 'PaymentType' = Postpaid, otherwise, it will not be sent.
4. 'BillingNo' element should be sent in payment inquiry request if 'PaymentType' = Postpaid.
5. 'BillingNo' element should be sent in payment inquiry request if 'PaymentType' = Prepaid and 'BillingNoRequired' = 'true'. If it is 'false', then, 'BillingNo' element should not be sent.
6. 'BillingNo' element will be sent in response if 'PaymentType' = Postpaid or if it is Prepaid and 'BillingNoRequired' = true.
7. 'ValidationCode' element should be sent in payment inquiry request if 'PaymentType' = Prepaid, otherwise, it should not be sent.
8. 'ValidationCode' element will be sent in response if 'PaymentType' = Prepaid, otherwise, it will not be sent.
9. If Error Code = '0' and payment message was processed successfully by ONEIC-TASDEED, then, 'ParFees' element will be sent in payment inquiry response.

Con 2 Us

Integration Specs

10. If Error Code = '0' and payment message was processed successfully by ONEIC-TASDEED, then, 'PmtStatus' element will be sent in payment inquiry response.
11. If Error Code = '0', then, ONEIC-TASDEED will send 'STMTDate' and 'JOEBPPSTrx' elements in response, otherwise, these elements will not be sent.
12. 'PrepaidCat' element should be sent in payment inquiry request if 'PaymentType' = Prepaid and 'ContainsPrepaidCats' = true, otherwise, it should not be sent.
13. 'PrepaidCat' element will be sent in response if 'PaymentType' = Prepaid and 'ContainsPrepaidCats' = true, otherwise, it will not be sent.
14. Payment Inquiry request should be only sent to ONEIC-TASDEED system if and only if the Banks/PSPs did not receive a bill payment/ prepaid payment response from ONEIC-TASDEED system.

6.19 Payment Notification

1. The 'ProcessDate' sent in Payment Notification request will equal the 'ProcessDate' received in its corresponding 'bill payment' / 'prepaid payment' response.
2. The 'Process Date' sent in Payment Notification response should equal the 'Process Date' sent in the corresponding request.
3. The 'STMTDate' sent in Payment Notification request will equal the 'STMTDate' sent in its corresponding 'bill payment' / 'prepaid payment' response.
4. The 'STMTDate' sent in Payment Notification response should equal the 'STMTDate' sent in the corresponding request.
5. If the bill contains Sub-Payments, then a Sub-Payments section will be sent in the corresponding Payment Notification request with all its details in case of 'Allow Partial' = false.
6. If the bill contains Sub-Payments, then a Sub-Payments section will be sent in the corresponding Payment Notification request with update the value of 'Amount' element to be equal the 'Paid Amount' in case of 'Allow Partial' = true.
7. If the biller has 'Allow multiple settlement banks'= true, then, the payment notification requests will be sent to both biller and settlement bank mentioned in his bills subpayments even if it is not the same settlement bank defined in biller profile in ONEIC-TASDEED.

C o n 2 U s

Integration Specs

8. If the biller has 'Allow multiple settlement banks'= true, and settlement bank mentioned in the bills sub-payments is not the same settlement bank defined in biller profile in ONEIC-TASDEED, then, no payment notification request will be sent to that bank for these bills.
9. If the biller has 'Allow multiple settlement banks'= false, then, the payment notification requests will be sent to both biller and biller's settlement bank mentioned in his bills sub-payments which will be the same settlement bank defined in biller profile in ONEIC-TASDEED.
10. In Payment Notification request, ONEIC-TASDEED will send Fees Amount = '0' if 'Fees on Biller' = false.
11. In Payment Notification request, ONEIC-TASDEED will send Fees Amount larger than '0' (most probably) if 'Fees on Biller' = true.
12. If 'PaymentType' element is sent in Payment Notification request, then, its value will be 'Postpaid' if it is related to Bill Payment transaction, and it will be 'Prepaid' if it is related to Prepaid Payment transaction.
13. If payment notification request is sent for a service with 'PaymentType'=Prepaid, and its 'BillingNoRequired' = false, then, the 'BillingNo' will not be sent in the request, otherwise, it will be sent.
14. If payment notification request is sent for a service with 'PaymentType'=Prepaid, then, ONEIC-TASDEED will send the 'ValidationCode' value of the prepaid transaction inside the 'BillNo' element.
15. 'BillerCode' element will be sent in request if 'ReqTyp' = 'BNKPMTNTFRQ' and will not be sent if 'ReqTyp' = 'BLRPMTNTFRQ'.
16. If ONEIC-TASDEED sent 'PmtSrc' element in payment notification request, then, it will have the value 'BNKPmt' if the corresponding payment transaction was initiated by a Bank, and will 'PSPPmt' if it was initiated by a PSP.
17. If payment notification request is sent for a service with 'PaymentType'=Prepaid and 'ContainsPrepaidCats' = true, then, 'PrepaidCat' element will be sent in the request, otherwise, it will not be sent.
18. The element 'BankCode' sent in payment notification request will contain the 'Paying Bank' code if the related payment is initiated by a Bank, and it will contain the 'PSP Settlement Bank' code if the related payment is initiated by a PSP.

6.20 Payment Acknowledgment

1. Payment Acknowledgment will be sent for:
 - 'PSP Settlement Banks' for the payments from its PSP(s) side and succeeded by ONEIC-TASDEED.
 - 'PSPs' for the payments from its side and succeeded by ONEIC-TASDEED.
 - 'Paying Banks' for the payments from its side and succeeded by ONEIC-TASDEED.
2. Banks/PSPs can either receive Payment Acknowledgment request online, and then, Banks/PSPs should respond online as well, or, they can download Payment Acknowledgment files per each settlement day from the standalone application.
3. ONEIC-TASDEED will send 'PmtSrc' = BNKPmt if the corresponding payment transaction was initiated by a Bank, and will send 'PmtSrc' = PSPPmt if it was initiated by a PSP.
4. 'BillNo' element will be sent in payment acknowledgment request if 'PaymentType' = Postpaid, and will not be sent if 'PaymentType' = Prepaid.
5. 'BillingNo' element will be sent in payment acknowledgment request if 'PaymentType' = Postpaid.
6. 'BillingNo' element will be sent in payment acknowledgment request if 'PaymentType' = Prepaid and 'BillingNoRequired' = 'true'. If it is 'false', then, 'BillingNo' element will not be sent.
7. 'ValidationCode' element will be sent in payment acknowledgment request if 'PaymentType' = Prepaid, otherwise, it will not be sent.
8. 'PrepaidCat' element will be sent in payment acknowledgment request if 'PaymentType' = Prepaid and 'ContainsPrepaidCats' = true, otherwise, it will not be sent.
9. 'PSPCode' element will be sent in request if the related payment request was initiated by a 'PSP' and if the receiver type of the acknowledgment request is 'PSP Settlement Bank', otherwise, it will not be sent.
10. 'FeesAmt' element will be sent in payment acknowledgment request if 'FeesOnBiller' = 'false' for the service, otherwise, it will not be sent.
11. 'ParFees' element in payment acknowledgment request represents the fees of the Paying Bank or PSP that initiated the payment request.

6.21 Biller Service Availability

1. All billers under ONEIC-TASDEED payment gateway should provide one billing number in case of post-paid to be used within the biller service availability to make sure that the billing system from biller side is up and running by sending a successful response to ONEIC-TASDEED system taking into consideration that any unsuccessful or timeout response will be considered as "biller is unavailable" for a specific period.
2. In case of all biller services were Up/Down, the message structure received from ONEIC-TASDEED system will contains only main container "Biller" with "Status" value "Up" / "Down".
3. In case one of biller services was Up/Down, the message structure received from ONEIC-TASDEED system will contains the main container "Biller" with "Status" value "Up" with its related services statuses.
4. The response from the Banks/PSPs should be True when the message received successfully and to be False in case of any internal failure from Banks/PSPs while processing the message.

6.22 System Service Availability

1. The response from the Banks/PSPs/Billers should be True when the message received successfully and to be False in case of any internal failure from Banks/PSPs/Billers while processing the message.
2. The "Period" Container will be an optional element and it might be used to send a more information about the down time of ONEIC-TASDEED system.

7. Appendices

7.1 Nationalities Codes and Descriptions

Country Name	Country Code
AC	Ascension Island
AD	Andorra
AE	United Arab Emirates
AF	Afghanistan

C o n 2 U s
Integration Specs

AG	Antigua and Barbuda
AI	Anguilla
AL	Albania
AM	Armenia
AN	Netherlands Antilles
AO	Angola
AQ	Antarctica
AR	Argentina
AS	American Samoa
AT	Austria
AU	Australia
AW	Aruba
AX	Aland Islands
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BD	Bangladesh
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BH	Bahrain
BI	Burundi
BJ	Benin
BM	Bermuda
BN	Brunei Darussalam
BO	Bolivia
BQ	Bonaire, Sint Eustatius and Saba
BR	Brazil
BS	Bahamas
BT	Bhutan
BV	Bouvet Island
BW	Botswana
BY	Belarus
BZ	Belize
CA	Canada
CC	Cocos (Keeling) Islands
CD	Congo, Democratic Republic

C o n 2 U s
 Integration Specs

CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Cote D'Ivoire (Ivory Coast)
CK	Cook Islands
CL	Chile
CM	Cameroon
CN	China
CO	Colombia
CR	Costa Rica
CS	Czechoslovakia (former)
CT	Canton and Enderbury Islands
CU	Cuba
CV	Cape Verde
CW	Curacao
CX	Christmas Island
CY	Cyprus
CZ	Czech Republic
DE	Germany
DJ	Djibouti
DK	Denmark
DM	Dominica
DO	Dominican Republic
DZ	Algeria
EC	Ecuador
EE	Estonia
EG	Egypt
EH	Western Sahara
ER	Eritrea
ES	Spain
ET	Ethiopia
FI	Finland
FJ	Fiji
FK	Falkland Islands (Malvinas)
FM	Micronesia
FO	Faroe Islands
FR	France
GA	Gabon

C o n 2 U s
 Integration Specs

GB	Great Britain (UK)
GD	Grenada
GE	Georgia
GF	French Guiana
GH	Ghana
GI	Gibraltar
GL	Greenland
GM	Gambia
GN	Guinea

GP	Guadeloupe
GQ	Equatorial Guinea
GR	Greece
GS	S. Georgia and S. Sandwich Isls.
GT	Guatemala
GU	Guam
GW	Guinea-Bissau
GY	Guyana
HK	Hong Kong
HM	Heard and McDonald Islands
HN	Honduras
HR	Croatia (Hrvatska)
HT	Haiti
HU	Hungary
HV	upper volta africa
ID	Indonesia
IE	Ireland
IL	Israel
IN	India
IO	British Indian Ocean Territory
IQ	Iraq
IR	Iran
IS	Iceland
IT	Italy
JM	Jamaica
JO	Jordan
JP	Japan
KE	Kenya

C o n 2 U s
 Integration Specs

KG	Kyrgyzstan
KH	Cambodia
KI	Kiribati
KM	Comoros
KN	Saint Kitts and Nevis
KP	Korea (North)
KR	Korea (South)
KW	Kuwait
KY	Cayman Islands
KZ	Kazakhstan
LA	Laos
LB	Lebanon
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
LY	Libya
MA	Morocco

MC	Monaco
MD	Moldova
MG	Madagascar
MH	Marshall Islands
MK	F.Y.R.O.M. (Macedonia)
ML	Mali
MM	Myanmar
MN	Mongolia
MO	Macau
MP	Northern Mariana Islands
MQ	Martinique
MR	Mauritania
MS	Montserrat
MT	Malta
MU	Mauritius

C o n 2 U s
 Integration Specs

MV	Maldives
MW	Malawi
MX	Mexico
MY	Malaysia
MZ	Mozambique
NA	Namibia
NC	New Caledonia
NE	Niger
NF	Norfolk Island
NG	Nigeria
NI	Nicaragua
NL	Netherlands
NO	Norway
NP	Nepal
NR	Nauru
NU	Niue
NZ	New Zealand (Aotearoa)
OM	Oman
PA	Panama
PE	Peru
PF	French Polynesia
PG	Papua New Guinea
PH	Philippines
PK	Pakistan
PL	Poland
PM	St. Pierre and Miquelon
PN	Pitcairn
PR	Puerto Rico
PS	Palestinian Territory, Occupied
PT	Portugal
PW	Palau
PY	Paraguay
PZ	Panama Canal Zone
QA	Qatar
RE	Reunion
RO	Romania

RU	Russian Federation
RW	Rwanda

C o n 2 U s
 Integration Specs

SA	Saudi Arabia
SB	Solomon Islands
SC	Seychelles
SD	Sudan
SE	Sweden
SG	Singapore
SH	St. Helena
SI	Slovenia
SJ	Svalbard & Jan Mayen Islands
SK	Slovak Republic
SL	Sierra Leone
SM	San Marino
SN	Senegal
SO	Somalia
SR	Suriname
ST	Sao Tome and Principe
SU	USSR (former)
SV	EI Salvador
SY	Syria
SZ	Swaziland
TC	Turks and Caicos Islands
TD	Chad
TF	French Southern Territories
TG	Togo
TH	Thailand
TJ	Tajikistan
TK	Tokelau
TL	Timor-Leste
TM	Turkmenistan
TN	Tunisia
TO	Tonga
TR	Turkey
TT	Trinidad and Tobago
TV	Tuvalu
TW	Taiwan
TZ	Tanzania
UA	Ukraine
UG	Uganda

C o n 2 U s
 Integration Specs

UM	US Minor Outlying Islands
US	United States
UZ	Uzbekistan
VA	Vatican City State (Holy See)
VC	Saint Vincent & the Grenadines
VE	Venezuela
VG	British Virgin Islands
VI	Virgin Islands (U.S.)
VN	Viet Nam
VU	Vanuatu
WF	Wallis and Futuna Islands
WS	Samoa
XK	Kosovo
YE	Yemen
YT	Mayotte
YU	Serbia and Montenegro (former Yugoslavia)
ZA	South Africa
ZM	Zambia
ZW	Zimbabwe