

Dynamic Accounting System

Application Development Interface

Version 4.1.3

Date: 2015.06.19

CONFIDENTIAL

Content

1. Overview	5
1.1 Http Operation	5
1.1.1 HTTP POST	5
1.1.2 URL Parameters	5
1.1.3 Info Page	6
1.2 API Options	7
1.2.1 API Services Options	7
1.2.2 API Product Options	8
1.2.3 API Style Product Options	8
1.2.4 API VIP Options	8
1.3 Authentication	9
1.3.1 Authentication by MD5	9
1.3.2 Overtime	9
1.4 Posting a Request	9
1.4.1 Request XML	9
1.4.2 Request parameters	10
1.5 Response	10
1.5.1 Response XML	10
1.5.2 Error Response XML	11
1.6 Data formats	12
1.7 List of API Services	12
1.8 SDK	13
1.9 Get Started	13
2. API Service: GetServerTime	15
2.1 Request	15
2.2 Response	15
3. API Service: Product Information	16
3.1 API Service: Product_GetInfoExact	16
3.1.1 Request	16
3.1.2 Request Parameters	16
3.1.3 Response	16
3.1.4 Fields	17
3.2 API Service: Product_GetInfo	20
3.2.1 Request	20
3.2.2 Request Parameters	20
3.2.3 Response	21

3.3 API Service: Product_Create/Product_Update	21
3.3.1 Request.....	21
3.3.2 Response	21
3.4 API Service: Style_GetInfo	21
3.4.1 Request.....	21
3.4.2 Response	22
4. VIP Info.....	23
4.1 VIP_GetInfo.....	23
4.1.1 Sample Request	23
4.1.2 Request Parameters.....	23
4.1.3 Sample Response.....	24
4.1.4 Fields.....	24
4.2 VIP_Create/VIP_Update	27
4.2.1 Request.....	27
4.2.2 Fields.....	27
4.2.3 Response	27
5. POS Transactions.....	28
5.1 POS Transactions Get.....	28
5.1.1 Sample Request	28
5.1.2 Request Parameters.....	28
5.1.3 Sample Response.....	29
5.1.4 Fields.....	30
5.2 POS Transactions Insert	31
5.2.1 Request.....	31
5.2.2 Fields.....	32
5.2.3 Response	32
5.3 POS Transactions Calculate.....	32
5.4 POS Transactions Hold	33
5.4.1 Request.....	33
5.4.2 Response	34
5.5 POS Transactions Hold Released.....	34
5.5.1 Request.....	34
5.5.2 Response	34
6. VIP Points Adjustment	35
6.1 VIP Points Adjustment Get.....	35
6.1.1 Sample Request	35
6.1.2 Request Parameters.....	35
6.1.3 Sample Response.....	35

- 7. Office Sales Transactions 36
 - 7.1 OfficeSaleTrans_Get 36
 - 7.1.1 Sample Request 36
 - 7.1.2 Request Parameters..... 36
 - 7.1.3 Sample Response..... 37
 - 7.1.4 Fields..... 38
 - 7.2 Office SaleTran_Insert 39
 - 7.2.1 Request..... 39
 - 7.2.2 Fields..... 40
 - 7.2.3 Response 40
- 8. Office Purchase Transactions 41
 - 8.1 API Service: PurchaseTrans_Get 41
 - 8.2 API Service: PurchaseTran_Insert 41

CONFIDENTIAL

1. Overview

1.1 Http Operation

1.1.1 HTTP POST

The API adopts the standard HTTP POST method. The URL is in the format of the following:

`http://[URL]/D_All/api?DATAID__[Data ID]&AUTH_TIME=[time in ms]&AUTH_ENCRYPT=[key]`

eg.

`http://192.168.1.123:8080/D_All/api?DATAID__=200_DATA&AUTH_TIME=12343&AUTH_ENCRYPT=34343434343`

1.1.2 URL Parameters

	Parameter	Description	Remarks
1.	DATAID__	The ID of the data to be handled	- this is the ID that appears in the banner of Dynamic Intranet System
2.	AUTH_TIME	The estimated server time	- This is required if Authentication is set as "MD5", see Options below - This time is for calculation the encryption key and for checking whether the request has already timed out - an API function is available for obtaining the server time
3.	AUTH_ENCRYPT	The encrypted value calculated based on a key set up by the server and known to the API software	- This is required if Authentication is set as "MD5", see Options below

1.1.3 Info Page

In the address bar of a browser, going to the URL the browser would show something like the following:

http://WebServerIPAddress:8080/D_All/api?DATAID_=DynamicDATAID

(A) Services supported

- This API function list interface shows the Services in API supported for the website

(B) Upgrade Notes

- Update notes of API Function for user's reference.

The screenshot shows a web browser window with the following content:

Dynamic API Version 4.11 2015.06.13 DataID: DEMOMOB

Services supported

- GetServerTime
- PingServer
- Product_GetInfo
- Product_Create
- Product_Update
- VIP_GetInfo
- VIP_Create
- VIP_Update
- POSTran_Hold
- POSTran_ReleaseHold
- OfficeSaleTran_Get
- OfficeSaleTran_Insert
- PurchaseTran_Get
- PurchaseTran_Insert
- POSTran_Calculate
- POSTran_Insert
- POSTran_Get
- VIPPoints_Adjust
- Style_GetInfo

Upgrade Notes

Date	Service	Description
15.06.13	Product_GetInfo	• Support product range: <PRODUCT1>,<PRODUCT2> / <COMPANY1>,<COMPANY2>. Can be used to get all records in batches
	Style_GetInfo	
	VIP_GetInfo	• Fields count and results count constraints from options
	VIP_GetInfo	• Search by <LASTUPDATE>. Can be used to get only records with changes since the date time.
	VIP_GetInfo	Search by exact code only option.

1.2 API Options

Dynamic provides flexibility to implementing the API by providing a number of options to control the use of the API functions. These options are available in the System options: API tab of Dynamic Intranet System.

1.2.1 API Services Options

API Functions of Dynamic Intranet System open for website provider.

API Options

Services	System RPC	Product	Style	VIP
RPC001	RPC Data Exchange:	<input type="text" value="....."/>		
API102	Authentication mode:	MD5 Encryption by time <input type="button" value="v"/>		
API103	Restricted API requester IPs:	<input type="text"/>		
	Calculate VIP Points while inserting:	<input type="checkbox"/>		
API201	Read product info:	All fields <input type="button" value="v"/>		
API202	Insert Product:	<input checked="" type="checkbox"/>		
API203	Update Product:	<input checked="" type="checkbox"/>		
API211	Read VIP Info:	<input checked="" type="checkbox"/>		
API212	Insert VIP Info:	<input checked="" type="checkbox"/>		
API213	Update VIP Info:	<input checked="" type="checkbox"/>		
API216	Insert VIP Adjustment:	<input checked="" type="checkbox"/>		
API221	Read POS transactions:	<input checked="" type="checkbox"/>		
API222	Insert POS transactions:	<input checked="" type="checkbox"/>		
API231	Read office sales:	<input checked="" type="checkbox"/>		
API232	Insert office sales:	<input checked="" type="checkbox"/>		
API241	Read purchase transactions:	<input checked="" type="checkbox"/>		
API242	Insert purchase transactions:	<input checked="" type="checkbox"/>		

Number	Caption	Remarks
1.	Authentication	<ul style="list-style-type: none"> - No: This is useful in the initial stage of development or IP restriction restricted is adopted - MD5: This is needed when the 3rd software may run at any computer
2.	Restricted IP	<ul style="list-style-type: none"> - Enter a fixed IP address so that other IP addresses are not accepted by the API services - If this option is left blank, there shall be no restrictions on which computer is to invoke the API services
3.	Release a service	<ul style="list-style-type: none"> - For each of the services supported, there is an option to control whether the service is to be released for use
4.	Calculation of VIP Points	<ul style="list-style-type: none"> - This option is applicable if the VIP Points module has been purchased

		<ul style="list-style-type: none"> - If yes, Dynamic will compute the VIP points in the way it does in Dynamic Standard POS - If no, the points need to be calculated prior to invoking the services
--	--	--

1.2.2 API Product Options

API Product Functions of Dynamic Intranet System restricted the number of search result and fields return to the website.

API Options				
Services	System RPC	Product	Style	VIP
Maximum search results:		100	▼	
Maximum return fields:		10	▼	

1.2.3 API Style Product Options

API Style Product Functions of Dynamic Intranet System restricted the number of search result and fields return to the website.

API Options				
Services	System RPC	Product	Style	VIP
Maximum search results:		100	▼	
Maximum return fields:		10	▼	
Maximum detail product return fields:		10	▼	

1.2.4 API VIP Options

API VIP Functions of Dynamic Intranet System restricted the number of search result and fields return to the website.

API Options				
Services	System RPC	Product	Style	VIP
Maximum search results:		10	▼	
Maximum return fields:		10	▼	
Search by VIP code only:		<input checked="" type="checkbox"/>		

1.3 Authentication

1.3.1 Authentication by MD5

- The user setup a password in the API options tab. The password needs NOT be included in the URL parameters or anywhere in the request XMLs
- A value using an MD5 algorithm based on the password and current time, is computed and sent as the AUTH_ENCRYPT parameter
- The value is validated at the server using the value of the AUTH_TIME parameter

1.3.2 Overtime

A toleration of 2 minutes is allowed for difference in time between the time of the client computer and the server time. Dynamic API provides a service to obtain the server time, which in turn may be used to emulate the server time at the API client..

1.4 Posting a Request

1.4.1 Request XML

A request is posted in XML format with UTF-8 encoding. Please refer to the sample request XMLs bundled with this documentation. An example of a request XML is as below:

```
<?xml version="1.0" encoding="utf-8"?>  
<Dynamic JOBID="VIP_GetInfo" REQUEST_ID="12345">  
<Company>12012</Company>  
<CARDNO></CARDNO>  
</Dynamic>
```

In the above the tags and values in bold are common to all requests

1.4.2 Request parameters

No.	Element/Attribute	Description	Remarks
1.	JOBID	An ID for the service defined by Dynamic	- for easy references, the parameters s are defined in APIParams.java contained in the Java API Samples
2.	REQUEST_ID	An ID unique to all requests	- Dynamic API would not accept a request ID to be used a second time - This request ID will appear in the API usage log
3	Other parameters	Other parameters are specific to individual API functions	- Please refer to the references to the detail services

1.5 Response

1.5.1 Response XML

The response to a request is written in XML format with UTF-8 encoding. Please refer to the sample response XMLs bundled with this documentation. An example of a response XML is as below:

```
<?xml version="1.0" encoding="utf-8"?>
<Response JOBID="VIP_GetInfo" REQUEST_ID="12345" HANDLED_REF="0000001">
<VIPInfo>
  <COMPANY>12345</COMPANY>
  <DESC_ENG>Mr Wong</DESC_ENG>
  <TELEPHONE>13212345678</TELEPHONE>
  ...
  ...
</VIPInfo>
</Response>
```

The attribute HANDLED_REF is a 7-digit serial number generated by Dynamic API and is shown in the API invocation log.

1.5.2 Error Response XML

In the case of an error, the <Response> element will contain the attribute “ERROR=[n];[Description]”. An example is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<Response JOBID="Product_GetInfo" REQUEST_ID="12346" ERROR="11;Product not found: DBM-XXX"
HANDLED_REF="0000123">
</Response>
```

Errors codes:

These errors are defined in the APIParams.java contained in the Java Samples.

No.	Error code	Description	Remarks
1.	0	Error not classified	Please refer to the accompanying description
2.	1	Over time	Request is not within tolerated range of the time
3.	2	Not authenticated	Encrypted key is not authenticated by the server
4.	3	Not permitted client	Client's ip address is not accepted by the deployment's restricted IP
5.	4	Service is not supported	
6.	10	Cannot parse request XML	
7.	11	Invalid date format	Date needs to be in yyyy.MM.dd format, eg. 2015.03.15. Examples of invalid formats are 03/15/2015, 15/03/2015, 2015-03-15
8.	20	Record not found	
9.	21	Record already exists	
10.	22	Detail search conditions are required	
11.	99	Unhandled error	Please report this error to Dynamic Software Development Ltd

1.6 Data formats

All dates are expressed in YYYY.MM.DD formats, eg. 2015.03.13. Date formats such as 03/13/2015, 13/03/2015 are treated as errors. All numbers are unformatted texts.

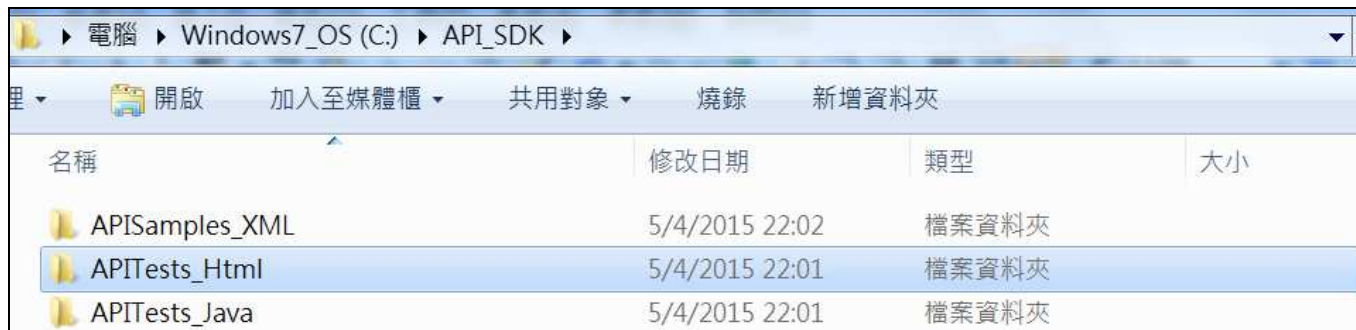
1.7 List of API Services

Note that there is an option to release each of the following service. Whether a service is supported may be scanned from API info page.

No.	API Services	API Job ID	Description
1.	System	PingServer	Ping server connection
		GetServerTime	Get server time
2.	Product	Product_GetInfoExact	Get one product information
		Product_GetInfo	Get a range of product information
		Product_Create	Insert new product
		Product_Update	Update existing product information
		Style_GetInfo	Get style product information
3.	VIP	VIP_GetInfo	Get VIP information
		VIP_Create	Insert new VIP
		VIP_Update	Update existing VIP information
4.	POS Transaction	POSTrans_Get	Get POS transaction information
		POSTran_Insert	Insert new POS transaction
		POSTran_Calculate	POS transaction item scheme calculation
		POSTran_Hold	POS transaction hold VIP Points
		POSTran_ReleaseHold	POS transaction release held VIP Points
5.	VIP Points adjustment	VIPPoints_Adjust	Insert VIP points adjustment
6.	Office Sales	OfficeSaleTrans_Get	Get office sales transaction information
		OfficeSaleTran_Insert	Insert office sales transaction
7.	Office Purchases	PurchaseTrans_Get	Get office purchases transaction information
		PurchaseTran_Insert	Insert office purchases transaction

1.8 SDK

An SDK is provided to the 3rd party software developer. The SDK comprises the following:



No.	Description	Remarks
1.	Documentation	This document
2.	Simple API html pages	<ul style="list-style-type: none"> - Contains a folder of html files and javascript files - Contains samples for a few services only - Run under MS IE only
3.	Sample API request and response XML files	<ul style="list-style-type: none"> - Samples per service
4.	Complete Java samples	<ul style="list-style-type: none"> - Runs under Java with a sample per each API service - The samples contains the file APIParams.java which define the constants used by Dynamic API

1.9 Get Started

No.	Step	Remarks
1.	Take a look at the sample XMLs	<ul style="list-style-type: none"> - Take a look at the sample request and response XMLs provided - For better effects associate any one of the XML files with a browser or a dedicated XML viewer
2.	Ascertain the api URL	<ul style="list-style-type: none"> - change the URL to the intranet URL and substitute /main to /api - In the address bar of a browser input [URL]?DATAID__[DataID] and confirm that the info message appears
3.	Configure the options	<ul style="list-style-type: none"> - Under the System Shell in the Setup/API menu of Dynamic Intranet select the services that are needed - Refresh the api info page and confirm that the selected services are listed

4.	Start with the no authentication mode	<ul style="list-style-type: none"> - turn off the encryption mode - leave the restricted IP value blank
5.	Try out the html samples	<ul style="list-style-type: none"> - Associate the html samples with the Microsoft Internet Explorer - Under the File Explorer click the html files - For convenience, the parameters contained in the params.js may be modified
6.	Try out the restricted IP feature	<ul style="list-style-type: none"> - Enter an IP not the same as that of the invoking computer - Execute the html samples at the invoking computer and take note of the error - Change the restricted IP address to that of the invoking computer - Execute the html samples again and note that the pages can now run
7.	Try out the java samples	<ul style="list-style-type: none"> - Change the URL setting in the APITestUtilities.getURL() in the same way as above - Run each of the samples and inspect the response XMLs written into the C:/dynamicTemp/[service]/ folder
8.	Try out the authentication mode	<ul style="list-style-type: none"> - Turn the authentication mode to MD5 - Enter a password into the key field - Write an incorrect value into the RPCKEY parameter in the same getURL() method as above - Run Product_GetInfo, or any one of the services that have been released for use, and inspect the response XML and note the failure - Change the RPCKEY parameter to the correct value and run the test again and note the success

2. API Service: GetServerTime

This service is useful if the client computer has to synchronize the time with the Intranet Server. It is necessary if MD5 encryption is adopted.

This service is the only service of Dynamic API that does not require a REQUEST_ID and does not return a HANDLED_REF.

2.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Dynamic JOBID="GetServerTime">
</Dynamic>
```

2.2 Response

```
<?xml version="1.0" encoding="utf-8"?>
<Dynamic JOBID="GetServerTime">
<ServerTime>1428550519666</ServerTime>
</Dynamic>
```

3. API Service: Product Information

3.1 API Service: Product_GetInfoExact

3.1.1 Request

```
<?xml version="1.0" encoding="utf-8" ?>
  <Dynamic JOBID="Product_GetInfo" REQUEST_ID="I1434701880">
    <PRODUCT1>DBM-FOX-PR2</PRODUCT1>
    <PRODUCT2>DBM-FOX-PR2</PRODUCT2>
  </Dynamic>
```

3.1.2 Request Parameters

No.	Element/Attribute	Description	Remarks
1.	PRODUCT	Product Code	Exact Search product by product code

3.1.3 Response

The result only shows one product record.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <Response JOBID="Product_GetInfo" REQUEST_ID="I1434701880" HANDLED_REF="0000029">
    <SearchSummary FIELDSCOUNT="78" DATABASECOUNT="1" LASTUPDATED="1427269713440"
    RETRIEVEDCOUNT="1" FIRSTKEY="DBM-FOX-PR2" LASTKEY="DBM-FOX-PR2" NEXTKEY=""
    TIMEELAPSED="111 ms" />
    <ProductInfo>
      <PRODUCT>DBM-FOX-PR2</PRODUCT>
      <DESC_ENG>Foxware Software V2.0</DESC_ENG>
      <DESC_CHI>Foxware 軟件 2.0 視窗版本</DESC_CHI>
      <UNITL>set</UNITL>
      <CHUNITL>套</CHUNITL>
      <BARCODE>4901345473524</BARCODE>
```



```

<F_GROUP>B</F_GROUP>
<VENDOR>APP</VENDOR>
<BRAND>B2</BRAND>
<MODEL>M1</MODEL>
<PRD_ORIGIN>CH</PRD_ORIGIN>
<FIRSTDATE>1900.01.01</FIRSTDATE>
<ST_COST>120.0</ST_COST>
<PRICEA>240.0</PRICEA>
<STOCK1>669.0</STOCK1>
<STOCK01>433.0</STOCK01>
...
...
</ProductInfo>
</Response>
    
```

3.1.4 Fields

No.	Field	Type	Length	Remarks
* 1.	PRODUCT	CHAR	14	- Product Code - Extend Width: 25 (Max)
2.	DESC_ENG	CHAR	25	- Product English Description 1 - Extend Width: 40 (Max)
3.	DESCRIP2	CHAR	25	- Product English Description 2 - Extend Width: 40 (Max)
4.	DESCRIP3	CHAR	25	- Product English Description 3 - Extend Width: 40 (Max) - Extended Option
5.	DESCRIP4	CHAR	25	- Product English Description 4 - Extend Width: 40 (Max) - Extended Option
6.	DESC_CHI	CHAR	25	- Product Chinese Description 1 - Extend Width: 40 (Max)
7.	CHDESCRIP2	CHAR	25	- Product Chinese Description 2 - Extend Width: 40 (Max)
8.	CHDESCRIP3	CHAR	25	- Product Chinese Description 3 - Extend Width: 40 (Max) - Extended Option
9.	CHDESCRIP4	CHAR	25	- Product Chinese Description 4

					- Extend Width: 40 (Max) - Extended Option
	10.	UNITL	CHAR	5	- Unit (English) - Extend Width: 7 (Max)
	11.	CHUNITL	CHAR	5	- Unit (Chinese) - Extend Width: 7 (Max)
	12.	UNITSCALE	NUMERIC		- For dual unit only
	13.	UNITSMALL	CHAR	6	- For dual unit only
	14.	CHUNITS	CHAR	6	- For dual unit only
	15.	QTY_TYPE	CHAR	1	- For dual unit only
	16.	BARCODE	CHAR	14	- Product Barcode - Account Group
	17.	BARCODE2,3	CHAR	14	- Extended Option
*	18.	F_GROUP	CHAR	3	- Account Group - Extend Width: 6 (Max)
	19.	SNID_MODE	CHAR	1	- If SNID Product, mark as 'F'
	20.	VENDOR	CHAR	9	- Vendor Code
	21.	BRAND	CHAR	8	- Brand Code
	22.	MODEL	CHAR	5	- Model Code
	23.	PRD_ORIGIN	CHAR	3	- Origin Code
	24.	FIRSTDATE	DATE		- First date
	25.	ST_COST	NUMERIC		- Standard Cost
	26.	PRICEA	NUMERIC		- Product Price A
	27.	PRICEB	NUMERIC		- Product Price B
	28.	PRICEC	NUMERIC		- Product Price C
	29.	PRICED,E,F,G,H	NUMERIC		- Product Price D to H - Extended Option
	30.	DISCOUNTA	NUMERIC		- Product Discount A
	31.	DISCOUNTB	NUMERIC		- Product Discount B
	32.	DISCOUNTC	NUMERIC		- Product Discount C
	33.	DISCOUNTD,E,F,G,H	NUMERIC		- Product Discount D to H - Extended Option
	34.	MINIMUM	NUMERIC		- Safe Quantity
	35.	REORDER	NUMERIC		- Reorder Quantity
	36.	SHELF	CHAR	5	- Shelf Code
	37.	WARN_MESS	CHAR	30	- Warning Message
	38.	WARN_TAG	CHAR	1	- Show Warning mark as 'Y'

39.	F_MESSAGE	CHAR	14	- Linked Message Code
40.	PACKING	CHAR	15	- Product Packing
41.	USE_TYPE	CHAR	1	- Type: Y: Normal N: Inactive P: Promotion T: Suspend
42.	VIP_PT_PER	NUMERIC		- Product VIP Points ratio - VIP module only
43.	SORDER_MIN	NUMERIC		- Minimum Sales Order Qty
44.	STOCK1	NUMERIC		- Total Stock Level - Updated by Dynamic Accounting System only
45.	STOCK01,02,03...	NUMERIC		- Location Stock Level - Updated by Dynamic Accounting System only
46.	SAFETY01,02,03 ...	NUMERIC		- Location Safety Qty - Extended option
47.	ISBN	CHAR	13	- Book ISBN number
48.	CAT_NO	CHAR	15	- Book Catalogue No.
49.	AUTHOR	CHAR	35	- Book Author
50.	TRANSLATOR	CHAR	35	- Book Translator
51.	PUBLISHER	CHAR	25	- Book Publisher
52.	BOOK_ORGPR	CHAR	12	- Book Original Price
53.	BOOK_DATE	DATE		- Book Published Date
54.	BOOK_PAGES	INT	4	- Book Page No.
55.	BOOK_CUT	INT	2	- Book Cutting
56.	BOOK_COVER	CHAR	1	- Book Cover : Blank = Paper Back, H = Hard Cover, S = Soft Cover
57.	BOOK_TYPE	CHAR	1	- Book Type : Blank = Book, P = Periodical
58.	BK_REVDATE	DATE		- Book Revised Date
59.	BK_VERSION	CHAR	2	- Book Version
60.	BK_LANG	CHAR	3	- Book Language
61.	DATETIME1	DT		- Web Shop Product Start Selling Date Time
62.	DATETIME2	DT		- Web Shop Product End Selling Date Time
63.	PICKUP_LNS	CHAR	10	- Product Pick up Locations
64.	BLOCK1	CHAR	3072	- Product Web Shop information Text Box 1

					E.g. Book Content Introduction
65.	BLOCK2	CHAR	3072		- Product Web Shop information Text Box 2 E.g. Book Author Introduction
66.	BLOCK3	CHAR	3072		- Product Web Shop information Text Box 3 E.g. Book Other Introduction
67.	IMAGE1,2,3	CHAR	40		- Product Web Shop Image Path
68.	SORTNUM	CHAR	4		- Product Web Shop Display Sort Number

In the above, fields marked with “*” is required field when add a new product record to Dynamic System.

3.2 API Service: Product_GetInfo

3.2.1 Request

Refer to Section 3.1.1.

3.2.2 Request Parameters

The result may shows more than one product records.

No.	Element/Attribute	Description	Remarks
1.	PRODUCT1	Product Code	From Product Code
2.	PRODUCT2	Product Code	To Product Code
3.	LASTUPDATE	Lastupate Product	Get the changed product
4.	BARCODE	Barcode	Dynamic API will try to match barcode, barcode2, barcode3 fields with the value of this parameter
5.	STYLEPRD	Style Prefix	
6.	DESCRIP	Product Description	
7.	F_GROUP	Account Group Code	
8.	BRAND	Brand Code	
9.	COLOR	Color Code	
10.	SIZE	Size Code	
11.	ISBN	Book ISBN	
12.	FIELDS_LIST	List out field	No. of field to be listed out subject to the restriction in Section 1.2.2 API Product Options

3.2.3 Response

The result may show more product records. Refer to 3.1.3 Response

3.3 API Service: Product_Create/Product_Update

3.3.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Dynamic JOBID=Product_Create REQUEST_ID='100012' >
<PRODUCT>XYZ-123-ABC</PRODUCT>
<BARCODE>800001/BARCODE>
<DESC_ENG>Wooden comb</DESC_ENG>
...
</Dynamic>
```

3.3.2 Response

```
<?xml version="1.0" encoding="utf-8"?>
<Response JOBID='Product_Create' REQUEST_ID='1023434' >
<PRODUCT>XYZ-123-ABC</PRODUCT>
</Response>
```

3.4 API Service: Style_GetInfo

3.4.1 Request

```
<?xml version="1.0" encoding="utf-8" ?>
<Dynamic JOBID="Style_GetInfo" REQUEST_ID="I1433996684">
```

```

<FIELDS_LIST>PRODUCT,DESC_ENG,PRICEA</FIELDS_LIST>
<PRODUCT_FIELDS_LIST>PRODUCT,STOCK1,STOCK01</PRODUCT_FIELDS_LIST>
<PRODUCT>SHI</PRODUCT>
<DESCRIP />
<F_GROUP />
<MODEL />
<BRAND />
</Dynamic>

```

3.4.2 Response

```

<?xml version="1.0" encoding="UTF-8" ?>
<Response JOBID="Style_GetInfo" REQUEST_ID="I1433996684" HANDLED_REF="0000021">
  <StyleInfo>
    <PRODUCT>SHI</PRODUCT>
    <DESC_ENG>Shirt</DESC_ENG>
    <PRICEA>100.0</PRICEA>
    <ProductInfo>
      <PRODUCT>SHI-BLK-L</PRODUCT>
      <STOCK1>233.0</STOCK1>
      <STOCK01>70.0</STOCK01>
    </ProductInfo>
    <ProductInfo>
      <PRODUCT>SHI-BLK-M</PRODUCT>
      <STOCK1>235.0</STOCK1>
      <STOCK01>70.0</STOCK01>
    </ProductInfo>
    <ProductInfo>
      <PRODUCT>SHI-BLK-S</PRODUCT>
      <STOCK1>235.0</STOCK1>
      <STOCK01>70.0</STOCK01>
    </ProductInfo>
    ...
  </StyleInfo>
</Response>

```

4. VIP Info

4.1 VIP_GetInfo

4.1.1 Sample Request

```
<?xml version="1.0" encoding="utf-8"?>
<Dynamic JOBID="VIP_GetInfo" REQUEST_ID="100010">
<Company></Company>
<CARDNO>3434343332</CARDNO>
<TELEPHONE></TELEPHONE>
</Dynamic>
```

4.1.2 Request Parameters

No.	Element/Attribute	Description	Remarks
1.	COMPANY	VIP Code	VIP Code
2.	CARDNO1	Card Number	VIP Card Number
3.	TELEPHONE	Telephone	Dynamic API will try to match the parameter value with the 3 telephone fields in the VIP info
4.	E_MAIL	Email	VIP Email address
5.	WECHAT	Wechat	VIP Wechat account
6.	FACEBOOK	Facebook	VIP Facebook account

4.1.3 Sample Response

Single or multiple records may be returned.

```
<?xml version="1.0" encoding="utf-8"?>
<Response JOBID="VIP_GetInfo" REQUEST_ID="123456" HANDLED_REF="0000124">
<Result>
<VIPInfo>
  <Company>123456</Company>
  <CARDNO>23433123456</CARDNO>
  <TELEPHONE>13228912022</TELEPHONE>
  <VIP_TYPE>A</VIP_TYPE>
  ...
  ...
</VIPInfo>
<VIPInfo>
  <Company>123457</Company>
  <TELEPHONE>13228912022</TELEPHONE>
  <CARDNO>23433123456</CARDNO>
  <VIP_TYPE>A</VIP_TYPE>
  ...
  ...
</VIPInfo>
</Dynamic>
```

4.1.4 Fields

No.	Field	Type	Length	Remarks	
*	1.	COMPANY	CHAR	9	- VIP Code
	2.	CARDNO1	CHAR	16	- VIP Card Number
	3.	DESC_ENG	CHAR	40	- English Name
	4.	DESC_CHI	CHAR	40	- Chinese Name
	5.	SEX	CHAR	1	- M or F
	6.	VIP_TITLE	CHAR	1	- VIP Title M: Mr 先生 F: Ms 女士 S: Mrs 太太 G: Miss小姐

					D: Dr 博士 P: Prof.教授 H: Hon.議員 1: 公公 2: 婆婆
	7.	VIP_PT_BAL	NUMERIC		- VIP Points Balance - VIP Points module only - This field is to be updated by Dynamic Accounting System
	8.	FIRSTDATE	DATE		- First Join Date
	9.	EXPIRYDATE	DATE		- VIP Expiry Date
	10.	BIRTHDATE	DATE		- VIP Birthday
	11.	TYPE	CHAR	1	- Mark "V" To indicate VIP
	12.	VIP_TYPE	CHAR	1	- VIP Group
	13.	VIP_LOC	CHAR	2	- VIP Join Location ID
	14.	TELEPHONE	CHAR	16	- Telephone Number 1
	15.	TELEPHONE2	CHAR	16	- Telephone Number 2
	16.	FAXNO	CHAR	16	- FAX Number
	17.	ADDRESS1	CHAR	40	- Address Line 1
	18.	ADDRESS2	CHAR	40	- Address Line 2
	19.	ADDRESS3	CHAR	40	- Address Line 3
	20.	ADDRESS4	CHAR	40	- Address Line 4
	21.	CAT	CHAR	1	- Price Category : A/B/C/D/E/F/G/H
	22.	MSG_TYPE	CHAR	1	- SMS Type: No : blank English SMS : '1' Chinese SMS : '2'
	23.	PROMO_TYPE	CHAR	1	- Promotion Type None: N SMS – English : 1 SMS – Chinese : 2 Email – English : 3 Email – Chinese : 4
	24.	VIA_MAIL	BOOLEAN	1	- Promotion via mail mark as 'Y'
	25.	VIA_EMAIL	BOOLEAN	1	- Promotion via email mark as 'Y'
	26.	VIA_TELE	BOOLEAN	1	- Promotion via telephone mark as 'Y'
	27.	VIA_SMS	BOOLEAN	1	- Promotion via SMS mark as 'Y'
	28.	VIA_W_APP	BOOLEAN	1	- Promotion via whatsapp mark as 'Y'

29.	VIA_WECHAT	BOOLEAN	1	- Promotion via wechat mark as 'Y'
30.	SORT1	CHAR	6	- Customer Group 1
31.	SORT2	CHAR	6	- Customer Group 2
32.	SORT3	CHAR	6	- Customer Group 3
33.	E_MAIL	CHAR	40	- Email Address
34.	SALESMAN	CHAR	10	- Salesman Code
35.	IDNO	CHAR	30	- ID number
36.	ISACTIVE	CHAR	1	- Customer Status Active : Y Inactive : N
37.	WARN_MESS	CHAR	35	- Customer Warning Message
38.	WARN_TAG	CHAR	1	- Mark 'Y' to indicate display Customer Warning Message
39.	REMARKS	CHAR	40	- Remarks
40.	CLIENTREF1	CHAR	15	- Reference Field 1
41.	CLIENTREF2	CHAR	25	- Reference Field 2
42.	FACEBOOK	CHAR	40	- Facebook address
43.	WEBSITE	CHAR	40	- Website
44.	STATE_TAG	CHAR	1	- Mark 'Y' to indicate print statement
45.	LABEL_TAG	CHAR	1	- Mark 'Y' to indicate print label
46.	TERM_MODE	CHAR	1	- Payment Term Mode: I: Invoice 發票 H: Cash 現金 M: Monthly 每月 L: L/C L/C O: Autopay 自動轉帳 X: Del. Date 交貨日期 Z: Del. Date 依交貨日期 Monthly 每月模式 T: T.T. T.T. C: C.B.T. C.B.T. D: D.P. D.P. A: D.A.D.P. P: P.D. 期票
47.	DUEDAY	DT		- Due day
48.	LIMIT	NUMERIC		- Credit limit

In the above, fields marked with "*" is required field when add a new VIP record to Dynamic System

4.2 VIP_Create/VIP_Update

4.2.1 Request

A sample request XML for VIP_Create is listed below:

```
<?xml version="1.0" encoding="utf-8"?>
<Dynamic JOBID="VIP_Create" REQUEST_ID="123457">
  <Company>12345</Company>
  <CARDNO></CARDNO>
  <DESC_ENG>Ms Wong</DESC_ENG>
  <DESC_CHI>黃女士</DESC_CHI>
  <TELEPHONE>13212345678</TELEPHONE>
  . . .
</Dynamic>
```

4.2.2 Fields

Please refer to Section 2.1.4 above. Note that the VIP_PT_BAL field is updated by Dynamic Accounting System.

4.2.3 Response

A sample response XML is listed below:

```
<?xml version="1.0" encoding="utf-8"?>
<Response JOBID="VIP_Create" REQUEST_ID="123457" HANDLED_REF="0000125">
  <COMPANY>123456</COMPANY>
</Response>
```

5. POS Transactions

5.1 POS Transactions Get

5.1.1 Sample Request

```
<?xml version="1.0" encoding="utf-8"?>
<Dynamic JOBID="POSTrans_Get" REQUEST_ID="123456">
  <TRANDATE>2015.04.04</TRANDATE>
  <SHOPNO></SHOPNO>
  <INVOICE></INVOICE>
  <COMPANY></COMPANY>
  <DELIVERED></DELIVERED>
</Dynamic>
```

5.1.2 Request Parameters

No.	Element/Attribute	Description	Remarks
1.	TRANDATE	Transaction Date	
2.	SHOPNO	Shop No or counter number	- A 2-digits code represents the location ignoring the counter - A 3-digits code represents a specific counter
3.	INVOICE	Invoice number	
4.	DELIVERED	Transaction Type	Y – Invoice N – Order R - Return
5.	COMPANY	VIP Code	

5.1.3 Sample Response

```
<?xml version="1.0" encoding="utf-8"?>
<Response JOBID="POSTrans_Get" REQUEST_ID="123457" HANDLED_REF="0000125">
<POSTran>
  <TranInfo>
    <INVOICE>123456</INVOICE>
    <COMPANY>012345</COMPANY>
    <SHOPNO>03</SHOPNO>
    <TRANDATE>2015.03.12</TRANDATE>
    ...
  </TranInfo>
  <EntryInfo>
    <PRODUCT>DBM-FOX-PR2</PRODUCT>
    <PRICE>120.0</PRICE>
    <QUANTITY>2</QUANTITY>
    <DISCOUNT>10</DISCOUNT>
    <SCHEME_NO>216</SCHEME_NO>
    <DESCRIP1></DESCRIP1>
  </EntryInfo>
  <EntryInfo>
    ...
  </EntryInfo>
</PosTran>
<PosTran>
  ...
  ...
</PosTran>
</Response>
```

5.1.4 Fields

No.	Field	Type	Length	Remarks
A.	TranInfo Fields			
A1.	INVOICE	CHAR	6	- Invoice Number
A2.	SHOPNO	CHAR	3	- Shop ID / Shop ID + Count ID - Example: 02 / 02A
A3.	TRANDATE	DATE		- Invoice Date
A4.	TRANTIME	CHAR	5	- in hh:mm format
A5.	REMARKS	CHAR	36	- Invoice Remarks
A6.	SALESMAN	CHAR	10	- Salesman Code
A7.	PAYTYPE1	NUMERIC	2	- Paytype No. (Payment method 1)
A8.	PAYAMT1	NUMERIC		- Payment Amount
A9.	REF1	CHAR	18	- Payment Reference
A10.	PAYTYPE2	NUMERIC	2	- Paytype No. (Payment method 2)
A11.	PAYAMT2	NUMERIC		- Payment Amount
A12.	REF2	CHAR	18	- Payment Reference
A13.	VIP_POINTS	NUMERIC		- Invoice Net Points (Equal to PTNS_ADD minus PTNS_MINUS) - VIP Points module only
A14.	PTNS_ADD	NUMERIC		- Invoice Points Gained - VIP Points module only
A15.	PTNS_MINUS	NUMERIC		- Invoice Points Deducted - VIP Points module only
A16.	VIP_AMOUNT	NUMERIC		- Invoice Calculated Points Amount - VIP Points module only
B.	EntryInfo Fields			
B1.	PRODUCT	CHAR	14	- Product Code - Width: Refer to Section 3.14 Product Field
B2.	DESCRIP1	CHAR	25	- Product Description - Width: Refer to Section 3.14 Product Field
B3.	QUANTITY	NUMERIC		- Product Quantity
B4.	PRICE	NUMERIC		- Product Unit Price
B5.	DISCOUNT	NUMERIC		- Product Discount - Example : 20% off , mark as "20.00"
B6.	AMOUNT	NUMERIC		- Product Total Amount

5.2 POS Transactions Insert

5.2.1 Request

Only 1 transaction may be inserted per request.

```
<?xml version="1.0" encoding="utf-8"?>
<Dynamic JOBID=POSTrans_Insert REQUEST_ID=123456>
<POSTran>
  <TranInfo>
    <INVOICE>AUTO</INVOICE>
    <DELIVERED>Y</DELIVERED>
    <COMPANY>012345</COMPANY>
    <SHOPNO>03</SHOPNO>
    <TRANDATE>2015.03.12</TRANDATE>
    ...
  </TranInfo>
  <EntryInfo>
    <PRODUCT>DBM-FOX-PR2</PRODUCT>
    <PRICE>120.0</PRICE>
    <QUANTITY>2</QUANTITY>
    <DISCOUNT>10</DISCOUNT>
    <SCHEME_NO>216</SCHEME_NO>
    <DESCRIP1></DESCRIP1>
  </EntryInfo>
  <EntryInfo>
    ...
  </EntryInfo>
</PosTran>
</Dynamic>
```

5.2.2 Fields

Please refer to 5.1.4 above.

5.2.3 Response

Only 1 transaction may be inserted per request.

```
<?xml version="1.0" encoding="utf-8"?>
<Response JOBID=POSTrans_Insert REQUEST_ID=123456>
<INVOICE>012343</INVOICE>
</Response>
```

5.3 POS Transactions Calculate

This function can calculate for product promotion scheme number, promotion price and discount to be in the POS transaction.

```
<?xml version='1.0' encoding='utf-8'?><Dynamic JOBID='POSTran_Calculate' REQUEST_ID='11434003014'><POSTran>
  <TranInfo>
    <INVOICE>AUTO</INVOICE>
    <COMPANY>00001</COMPANY>
    <SHOPNO>04</SHOPNO>
    <INVTYPE></INVTYPE>
    <TRANDATE>2015.01.23</TRANDATE>
    <TRANTIME>15:23</TRANTIME>
    <REMARKS>r...</REMARKS>
    <VIP_POINTS>0</VIP_POINTS>
    <PAYAMT1>222</PAYAMT1>
    <PAYTYPE1>2</PAYTYPE1>
    <REF1></REF1>
  </TranInfo>
  <EntryInfo>
    <PRODUCT>DBM-FOX-PR2</PRODUCT>
```



```

<PRICE>120</PRICE>
<QUANTITY>2</QUANTITY>
<DISCOUNT>10</DISCOUNT>
<AMOUNT>216</AMOUNT>
<DESCRIP1></DESCRIP1>
<SCHEME_NO></SCHEME_NO>
<PRICECAT></PRICECAT>
<SOURCE>01</SOURCE>
</EntryInfo>
<EntryInfo>
  <PRODUCT>*01S</PRODUCT>
  <AMOUNT>50</AMOUNT>
</EntryInfo>
</POSTran>
</Dynamic>

```

5.4 POS Transactions Hold

This function can hold VIP points before the POS transaction complete which avoided the double use of the points. System will auto release the points held after POS transaction inserted.

5.4.1 Request

```

<?xml version="1.0" encoding="utf-8" ?>
<Dynamic JOBID="POSTran_Hold" REQUEST_ID="I1434009522">
  <VIPPoints>
    <COMPANY>0300001</COMPANY>
    <VIP_POINTS>120</VIP_POINTS>
  </VIPPoints>
</Dynamic>

```

5.4.2 Response

```
<?xml version="1.0" encoding="UTF-8" ?>
  <Response JOBID="POSTran_Hold" REQUEST_ID="I1434009522" HANDLED_REF="0000022">
    <HeldInfo>
      <HOLD_ID>1434009525123-0</HOLD_ID>
      <COMPANY>0300001</COMPANY>
      <BALANCE_BEFORE>3945</BALANCE_BEFORE>
      <HELD>120</HELD>
      <BALANCE_AFTERH>3825</BALANCE_AFTERH>
    </HeldInfo>
  </Response>
```

5.5 POS Transactions Hold Released

This function can release the VIP points which held by Function 5.4 Transactions Hold. It is usually used for abandoning the transaction.

5.5.1 Request

```
<?xml version="1.0" encoding="utf-8" ?>
  <Dynamic JOBID="POSTran_ReleaseHold" REQUEST_ID="I1434011452">
    <HOLD_ID>I1434011438</HOLD_ID>
  </Dynamic>
```

5.5.2 Response

This function can refer to 5.4.2.

6. VIP Points Adjustment

6.1 VIP Points Adjustment Get

6.1.1 Sample Request

```
<?xml version="1.0" encoding="UTF-8"?>
<Dynamic REQUEST_ID="I1429166803470" JOBID="VIPPointsAdjust">
<TranInfo><COMPANY>00001</COMPANY>
<VIP_POINTS>110</VIP_POINTS>
<TRANDATE>2015.04.16</TRANDATE>
<F_REF1>r...</F_REF1>
</TranInfo>
</Dynamic>
```

6.1.2 Request Parameters

No.	Element/Attribute	Description	Remarks
1.	TRANDATE	Transaction Date	
2.	INVOICE	Invoice Number	
3.	COMPANY	VIP Code	

6.1.3 Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response HANDLED_REF="0000041" REQUEST_ID="I1429166803470" JOBID="VIPPointsAdjust">
<TRANNO>0000005</TRANNO>
</Response>
```

7. Office Sales Transactions

7.1 OfficeSaleTrans_Get

7.1.1 Sample Request

```
<?xml version="1.0" encoding="UTF-8"?>
-<Dynamic REQUEST_ID="I1429167744848" JOBID="OfficeSaleTrans_Get">
<TRANDATE>2015.04.16</TRANDATE>
<DELIVDATE/>
<INVOICE/>
<DISTBIN/>
<COMPANY/>
<DELIVERED>Y</DELIVERED>
<INVTYPE/>
<SALESMAN/>
</Dynamic>
```

7.1.2 Request Parameters

No.	Element/Attribute	Description	Remarks
1.	TRANDATE	Transaction Date	
2.	SHOPNO	Shop ID	- A 2-digits code represents the location - A 3-digits code represents a specific counter
3.	INVOICE	Invoice Number	
4.	DELIVERED	Invoice Type	Y – Invoice N – Order R - Return
5.	COMPANY	VIP Code	

7.1.3 Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
-<Response HANDLED_REF="0000045" REQUEST_ID="I1429167744848" JOBID="OfficeSaleTrans_Get">
<ORTran>
<TranInfo>
<INVOICE>00000001</INVOICE>
<INVTYPE/>
<TRANDATE>2015.04.16</TRANDATE>
<DELIVDATE>2015.04.16</DELIVDATE>
<TERM_MODE/>
<DUEDATE>2015.04.16</DUEDATE>
<DISTBIN>03</DISTBIN>
<REMARKS>REMARKS</REMARKS>
</TranInfo>
<EntryInfo>
<PRODUCT>DBM-FOX-PR2</PRODUCT>
<DESCRIP1>itemref</DESCRIP1>
<QUANTITY>2.0</QUANTITY>
<DISCOUNT>10.0</DISCOUNT>
<PRICE>120.0</PRICE>
<AMOUNT>216.0</AMOUNT>
</EntryInfo>
<EntryInfo>
...
...
</EntryInfo>
</ORTran>
</Response>
```

7.1.4 Fields

No.	Field	Type	Length	Caption	Remarks
A.	TranInfo Fields				
A1.	INVOICE	CHAR	6		- Invoice Number
A2.	COMPANY	CHAR	9		- Customer Code
A3.	DISTBIN	CHAR	2		- Location ID
A4.	TRANDATE	DATE			- Invoice Date
A5.	TRANTIME	CHAR	5		- in hh:mm format
A6.	DUEDATE	D			- Due day
A7.	DELIVDATE	D			- Delivery day
A8.	TERM_MODE	CHAR	2		- Payment Terms A - D.A. C - C.B.T. D - D.P. H - Cash I - Invoice L - L/C M - Monthly O - Autopay P - P.D. T - T.T. X - Delivery Date
A9.	INVTYPE	CHAR	3		- Invoice Type
A10.	OURREF	CHAR	12		- Your reference
A11.	YOURREF	CHAR	12		- Our reference
A12.	REMARKS	CHAR	35		- Remarks
A13.	SALESMAN	CHAR	10		- Salesman Code
A14.	CURRENCY	CHAR	3		- Currency of the Invoice
A15.	CURRATE	NUMERIC	8	6	- Currency exchange rate
B.	EntryInfo Fields				
B1.	PRODUCT	CHAR	14		- Product Code - Width: refer to Section 3.14 Product Field
B2.	DESCRIP1	CHAR	25		- Description 1 - Width: refer to Section 3.14 Product Field

B3.	QUANTITY	NUMERIC			- Product Quantity
B4.	PRICE	NUMERIC			- Product Unit Price
B5.	DISCOUNT	NUMERIC			- Product Discount
B6.	AMOUNT	NUMERIC			- Product Total Amount

7.2 Office SaleTran_Insert

7.2.1 Request

Only 1 transaction may be inserted per request.

```
<?xml version="1.0" encoding="UTF-8"?>
<Dynamic REQUEST_ID="I1429165946812" JOBID="OfficeSaleTran_Insert">
<Transaction>
<TranInfo>
<DELIVERED>Y</DELIVERED>
<INVOICE>AUTO</INVOICE>
<COMPANY>COMP1</COMPANY>
<DISTBIN>03</DISTBIN>
<TRANDATE>2015.04.16</TRANDATE>
<DUEDATE>2015.04.16</DUEDATE>
<DELIVDATE>2015.04.16</DELIVDATE>
<REMARKS>REMARKS</REMARKS>
</TranInfo>
<EntryInfo>
<PRODUCT>DBM-FOX-PR2</PRODUCT>
<PRICE>120</PRICE>
<QUANTITY>2</QUANTITY>
<DISCOUNT>10</DISCOUNT>
<AMOUNT>216</AMOUNT>
<DESCRIP1>itemref</DESCRIP1>
</EntryInfo>
<EntryInfo>
...
</EntryInfo>
</Transaction>
</Dynamic>
```

7.2.2 Fields

Please refer to 7.1.4 above.

7.2.3 Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response HANDLED_REF="0000021" REQUEST_ID="I1429165946812" JOBID="OfficeSaleTran_Insert">
<INVOICE>00000001</INVOICE>
</Response>
</PosTran>
<PosTran>
  ...
  ...
</PosTran>
</Response>
```

CONFIDENTIAL

8. Office Purchase Transactions

8.1 API Service: PurchaseTrans_Get

Similar to OfficeSaleTrans_Get. Please refer to Section 7.1 but change the JOBID to PurchaseTrans_Get

8.2 API Service: PurchaseTran_Insert

Similar to OfficeSaleTran_Insert. Please refer to Section 7.2 but change the JOBID to PurchaseTrans_Insert

CONFIDENTIAL