Oracle® MICROS Simphony First Edition

Transaction Services Overview and Troubleshooting
Version 1.8 and Later

November 2020

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Date	Version / Revision	Description/Reason for Changes
01/21/2011	1.0	Initial Version
10/20/2011	1.1	PostToTransaction
02/13/2012	1.5	GetTransaction request new parameters
02/28/2012	1.5	Added list of error code and user friendly message
03/09/2012	1.5	Added GetEmployee Extensions to GetConfigInfo Added new parameter EmployeeNumber to PostNewTransaction and PostToTransaction
03/30/2012	1.5	Added new Errors PreviousRoundItemFound PostingDiscountDetailNotSupported TipServiceChargeNotConfig
04/02/2012	1.5	Added new Errors InvalidAutoServiceCharge InvalidServiceChargeTaxClass
04/05/2012	1.5	Added new Errors InvalidTenderMedia ServiceTotalTenderMediaNotFound
04/06/2012	1.5	Added new Errors MissingTaxExemptReference Removed TenderTypeNotFound
05/02/2012	1.5	Added new Errors CreditCardPreambleNoMatchTender InitialCCAuthNotFound
05/22/2012	1.5	Added autofire information
06/13/2012	1.5	Added new Errors InvalidDeliveryTime MaximumAutofireChecksAllowed
07/11/2012	1.5	Check Journal Support
07/17/2012	1.5	Added NoDataFound error
08/07/2012	1.5	Added InvalidTenderMediaType error
8/30/2012	1.5	Added InvalidEGatewayConnectionURL InvalidLogZone
9/9/2013	1.5.501	Added MenuItemNotActiveOnMenuLevel
9/27/2013	1.6.0009	Added Order Type Order Device Routing support
4/11/2013	1.6.800.44	Updated Simphony Version Compatibility
7/2018	1.7	• Updated Chapter 10’s Limitations section • Updated the cover page and copyright pages
11/2019	1.7.x and later	Updated the release number on the title page
01/2020	1.7.x and later	Updated the product name on the cover page
11/2020	1.8.x and later	Updated the product version numbers throughout the doc

Table of Contents

1 Introduction

The Transaction Services application is a web service that allows third party applications to retrieve POS data definitions (tender media, menu items, etc.) and create checks within a POS. Transaction Services is designed to handle fast transaction environments, so it has little to no functionality that makes it useful within a table-service environment.

Other Documents

Document Name	Comments
Transaction Services Guide	This document describes installing Transaction Services.
Transaction Services WSDL Explanations Guide	This document describes the Data Access WSDL document supported by Transaction Services.
Data Access Service Guide	This document describes import and exporting data to and from the Simphony First Edition's database.

2 Terminology

Web Service:
This is the application providing some set of services to a client. These two entities communicate over HTTP using a well-defined language known as a WSDL. For the purposes of this document, Transaction Services is a *Web Service*.

Web Service Consumer (WSC):
This is the client application taking advantage of the services provided by a *Web Service*.

Web Service Description Language (WSDL):
This is an XML language that defines communication between a web service consumer (client) and a web service (server).

Data Access:
This is the component of Transaction Services that allows data definitions to be exported from the Simphony system. Examples of data definitions exported include menu items and tenders.

Web Ordering:
This is the component of Transaction Services that allows a third party application acting as a web service consumer to create checks within the Simphony First Edition system.

3 Security

Transaction Services does not have encryption or any other security mechanisms built into it. If a customer wishes to secure the Transaction Services web service, we advise that the customer configure SSL on the server. We offer no guidelines or assistance on SSL setup.

4 Licensing

Every WSC utilizing Transaction Services should be licensed. WSCs are licensed as workstation clients. Note that WSCs are configured as POSAPI or Workstation clients in EMC.

It is important to note that every WSC should be acting as a different POSAPI client. Some customers may attempt to use the same POSAPI client profile among several WSCs, but this is not a valid configuration. In such a configuration, the WSCs may experience ***random*** failures because Transaction Services cannot create checks using the same POSAPI client profile to open and close multiple checks at the same time.

Installation/Deployment

4.1 Location

Transaction Services needs to be installed separately from the Simphony application. It can be installed on the machine where WSC is running or on different machine/server. Each instance of Transaction service is linked to a workstation id, thus for each WSC, transaction services needs to be installed separately.

The location of the database server is of no consequence because all access to the database is funneled through the Simphony application.

4.2 Version Compatibility

Transaction Services must be installed with a compatible version of Simphony First Edition 1.x.

Compatible versions:

- Simphony First Edition version 1.6 MR8
- Simphony First Edition version 1.7.x
- Simphony First Edition version 1.8.x

4.3 Installation

Transaction Services can be installed using a setup file. The setup file creates a virtual directory TSWebservice on the selected website.

Run the setup file to install Transaction Services. Refer to the *Transaction Services Guide* for more information.

4.4 Initial Configuration

After Transaction Services has been installed, the web.config file may need to be updated. After updating the web.config file, run **TSConfig.exe** to generate a token id and the required registry keys in the registry.

> **Note:** Step 4.4 is only required if the PingServer method is not working after setup installation. See PingServer in chapter 5 for more information.

4.4.1 web.config file

The web.config file can be found in directory `C:\Inetpub\wwwroot\TSWebService`. The fields in the web.config file needs to be updated after Transaction service has been successfully installed. Some of the fields are populated by default. Below is the list of keys in this file.

- LogFileDirectory

This value in this contain indicates the directory where the log file will be created. A log file named *Transaction Services.log* will be created under this directory. This is filled with default value `C:\Logs`.

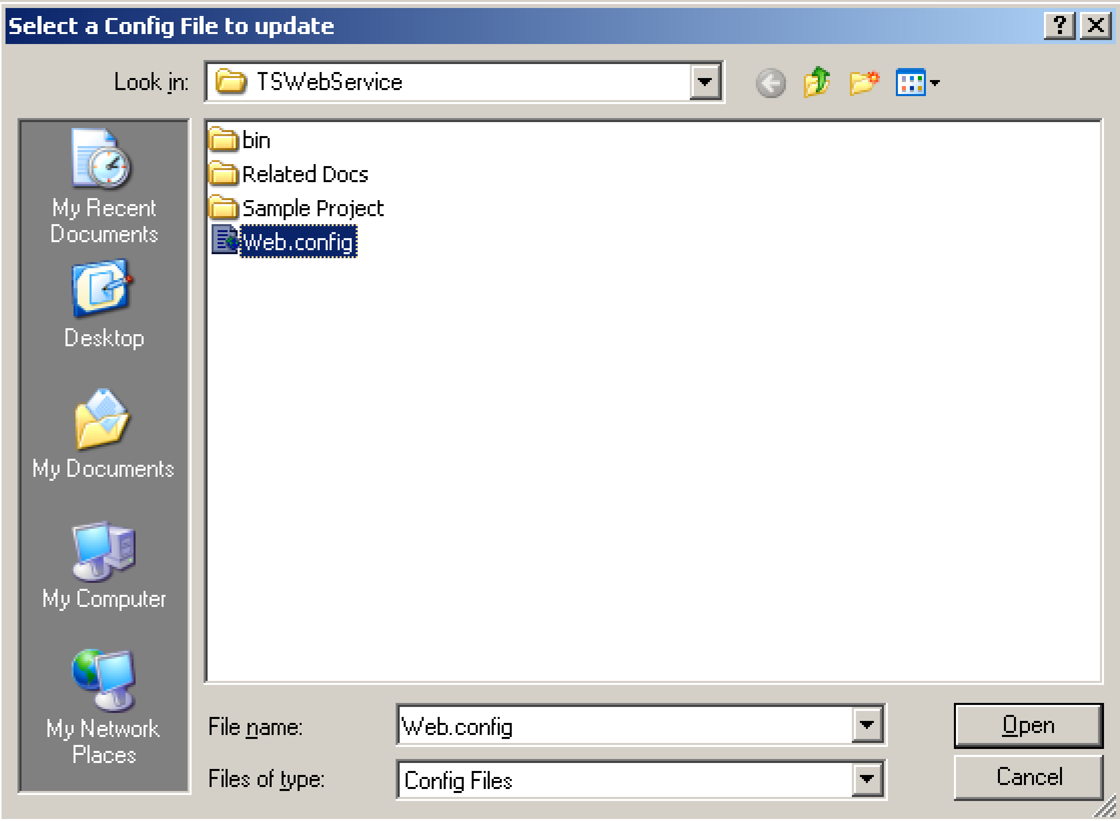
- TSWebService1_0
  This key indicates the verbosity level for logging in TSWebService zone.
- TSOps
  This key indicates the verbosity level for logging in TSOps zone.
- WSId
  This key indicates the workstation id. Change the value with the *Workstation ID* to be used. Workstation id can be obtained from EMC Workstation module.
- DefaultDBRefreshInterval
  This key indicates the default database refresh rate to download the datastore. It is filled with default value of *30 minutes* and can be changed as required.
- BaseDirectory
  This key indicates the base directory to download datastore. The default datastore location is *C:\TSDataStore* and can be changed as required.
- EmployeeObjId
  This key indicates the employee id. Change the value with the *Employee object number* to be used. Employee object number can be obtained from EMC Employee maintenance module.
- DefaultServiceTotalTender
  This key indicates the default tender media. Change the value with the *Tender media number* to be used. Tender media number can be obtained from EMC Tender Media module (Number field). Note that the default tender media number is used for internal processing only.
- ActiveHostIpAddress
  This key indicates the host url. Change the value with the *host EGateway url*
- DisplayMissingStringID
  This key is a Boolean option whether to display the string id value which could not be found in datastore or not.
- TokenId
  This key value pair gets generated by a tool *TSConfig.exe.* This field indicates that the client has been authenticated against server.

- SupportServiceTotal
  This key Boolean value is a control option to switch on or off for the permission to Post Transaction with or without payment.

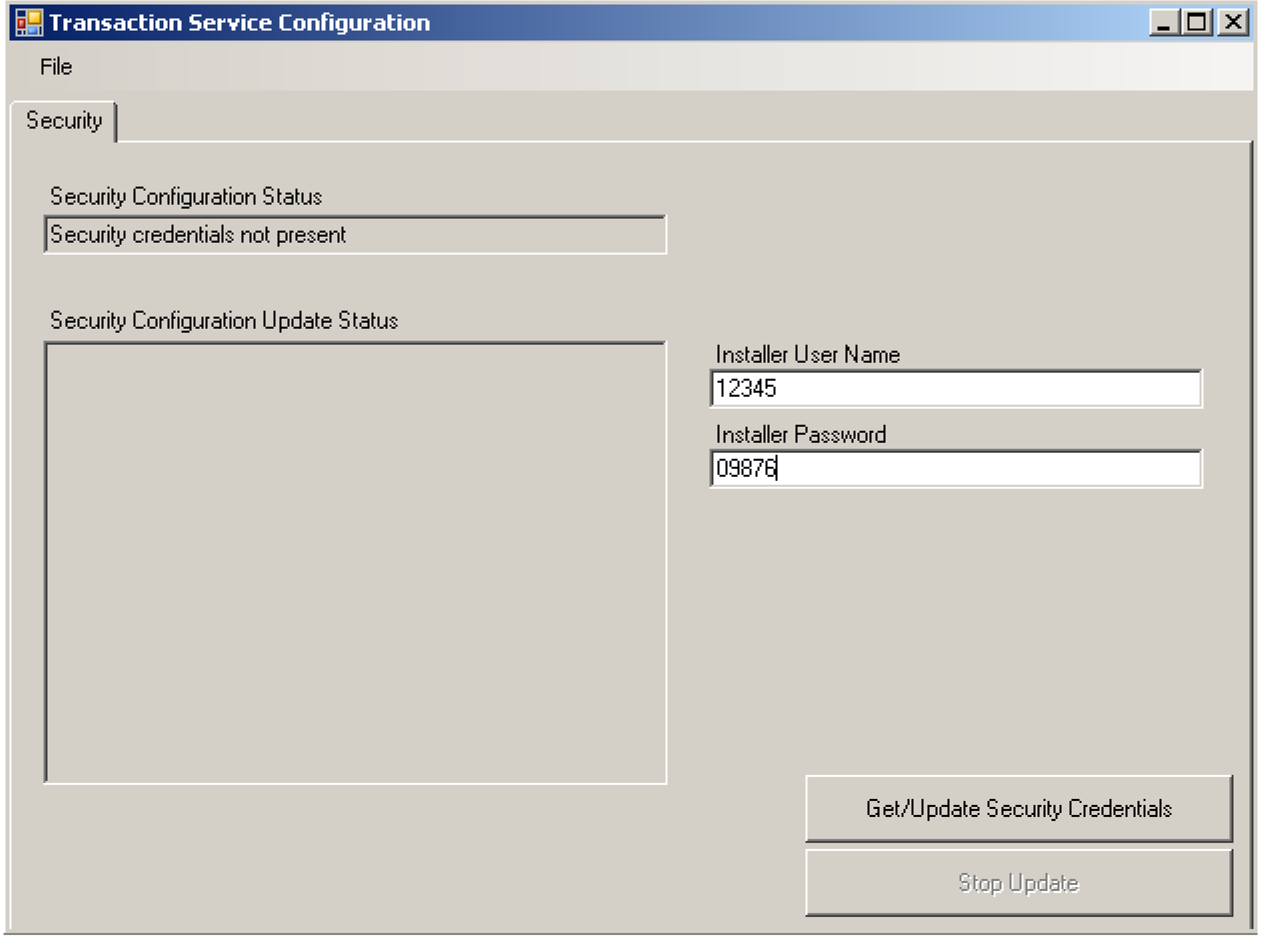
4.4.2 Authenticate client

The client needs to be authenticated using TSConfig.exe tool which can be found in directory `C:\inetpub\wwwroot\TSWebService\bin` tool.

Run TSConfig.exe. Select the web.config file in `C:\inetpub\wwwroot\TSWebService` directory.



On the Transaction Service Configuration window, enter the installer username and password and click Get/Update security credentials. The installer username and password can be obtained from EMC Property Parameters module.


Transaction Service Configuration
File
Security

Security Configuration Status

Security credentials not present

Security Configuration Update Status

Installer User Name

12345

Installer Password

09876

Get/Update Security Credentials
Stop Update

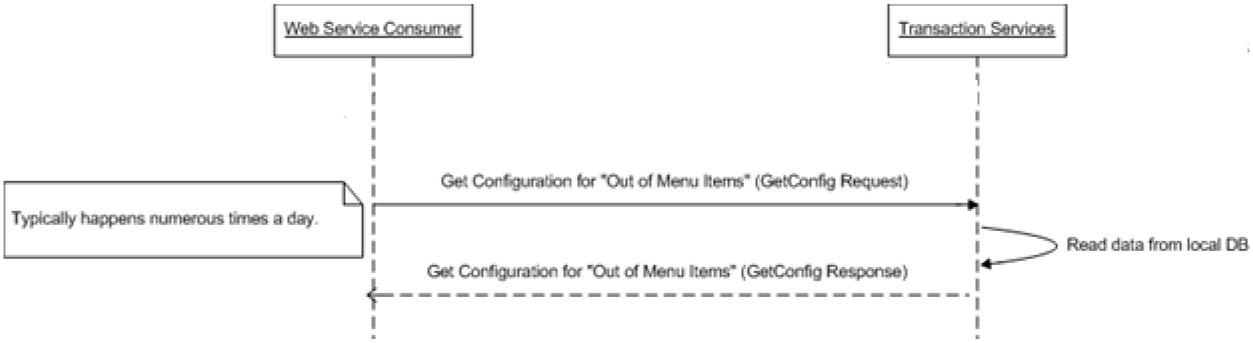
TSConfig.exe generates the token id field in the web.config file. This utility also creates the registry keys Userdata and UserdataXXX (XXX being workstation id) in: `HKLM\Software\MICROS`.

Transaction Services Components

4.5 Data Access

The Data Access portion of Transaction Services allows WSCs to retrieve record definitions from Simphony. Data Access retrieves information on menu items, modifiers, service charges, tenders, general parameters, order types, discounts and out of menu item.

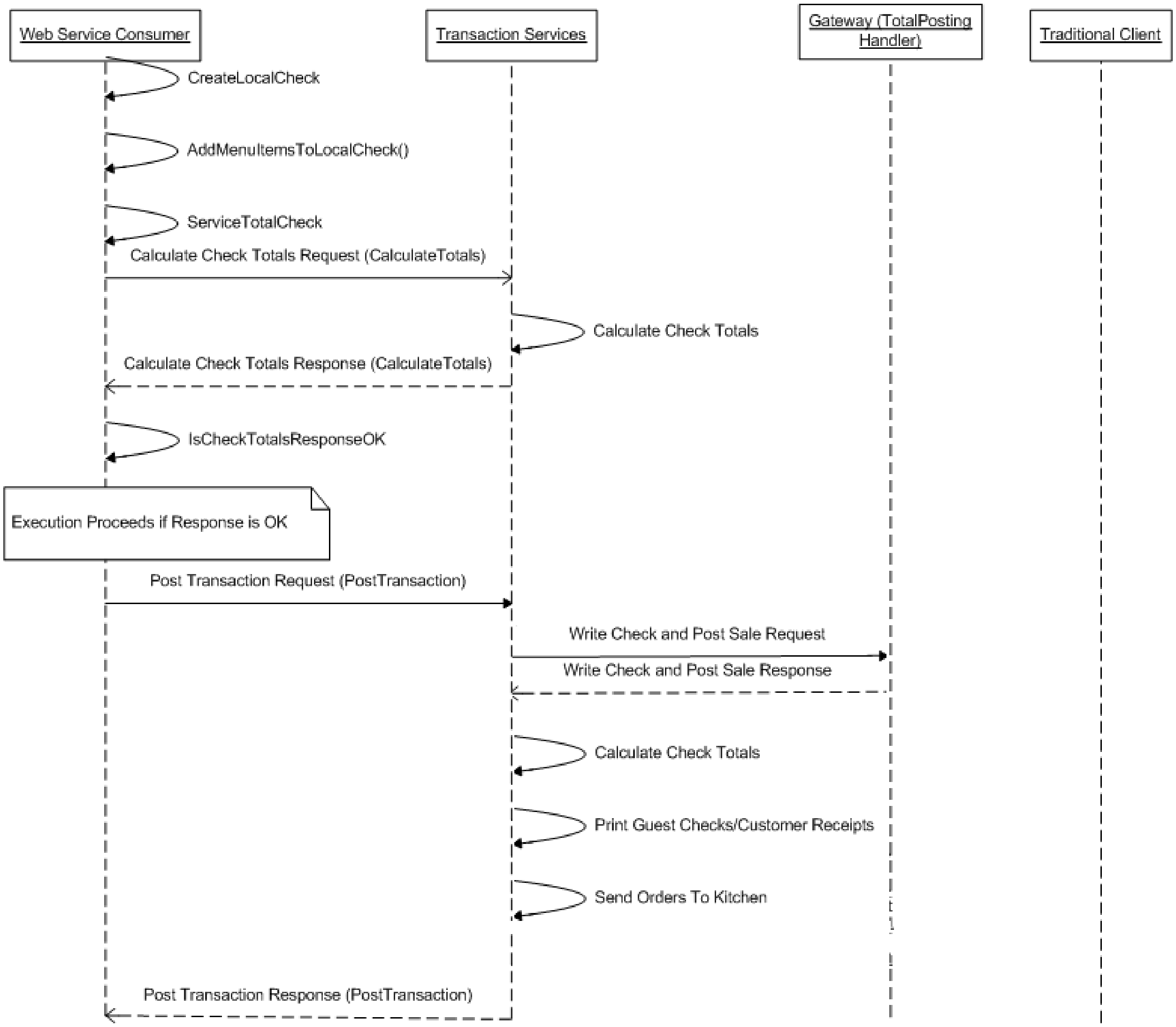
The following diagram shows the flow how a WSC can run for GetConfigInfo.


Web Service Consumer
Transaction Services
Typically happens numerous times a day.
Get Configuration for "Out of Menu Items" (GetConfig Request)
Read data from local DB
Get Configuration for "Out of Menu Items" (GetConfig Response)

4.6 Web Ordering

The Web Ordering portion of Transaction Services allows WSCs to calculate check totals, post transactions (create checks) within Simphony. Web Order can create open checks and closed checks, but it cannot pickup or split checks.

The following diagram does not dictate exactly how messages will flow back and forth between the Transaction Services Web Service and the web service consumer (i.e. a kiosk). It simply indicates the message flows that are *likely* to happen. The actions taken by Transaction Services as depicted in the diagram are in the correct order.


Web Service Consumer
Transaction Services
Gateway (TotalPosting Handler)
Traditional Client
CreateLocalCheck
AddMenuItemsToLocalCheck()
ServiceTotalCheck
Calculate Check Totals Request (CalculateTotals)
Calculate Check Totals
Calculate Check Totals Response (CalculateTotals)
IsCheckTotalsResponseOK
Execution Proceeds if Response is OK
Post Transaction Request (PostTransaction)
Write Check and Post Sale Request
Write Check and Post Sale Response
Calculate Check Totals
Print Guest Checks/Customer Receipts
Send Orders To Kitchen
Post Transaction Response (PostTransaction)

WSDL Overviews
Detail explanation of WSDL file can be found in the *Transaction Services WSDL Explanation* document.

4.7 Data Access

The methods that are exposed by Transaction Services are outlined in the *Data Access Service Guide* document. Here is a listing of the exposed methods:

4.7.1 GetConfigInfo:

GetConfigInfo returns Configuration Information based on Configuration request.

Configuration information contains

1. Modifier (Condiment) Definitions
   Retrieves modifier definitions for a given RVC within Simphony

2. Menu Item Definitions
   Retrieves menu item configuration for a given RVC within Simphony

3. Out of Menu Item Definitions
   Retrieves out of menu item configuration for a given RVC within Simphony

4. Menu Item Class
   Retrieves menu item class configuration from Simphony

5. Menu Item Price
   Retrieves menu item price configuration from Simphony

6. Menu Item SLU
   Retrieves menu item screen lookup configuration from Simphony

7. Menu Levels
   Retrieves menu item level configuration from Simphony

8. Service Charge Definitions
   Retrieves service charge configuration from Simphony

9. Discount Promotion Definitions
   Retrieves Discount configuration from Simphony

10. Tender Definition
    Retrieves tender media configuration from Simphony

11. Order Type Definitions
    Retrieves order type configuration from Simphony
12. Language Definitions
    Retrieves Language configuration from Simphony
13. General Parameter Definitions.
    Retrieves general parameter configuration from Simphony
14. Revenue Centers
    Retrieves revenue centers configuration from Simphony
15. Family Group
    Retrieves family group configuration from Simphony
16. Major Group
    Retrieves major group configuration from Simphony
17. Employee
    Retrieves employee(s) information from Simphony

4.7.2 PingServer

PingServer returns the state of the EGateway and the database server. This method can be used if the server is up and running, or not.

4.7.3 GetTransaction

This message retrieves a list of open or closed checks for a given set of parameters.
If optional parameter check GUID is specific, this message will retrieve the specific check with the given check GUID.

If check GUID is provided with other search criteria such as table number or employee number, but the matching check does not have the same information as the search criteria, data mismatch errors will be returned in the response message.

4.8 Web Ordering

The Web Ordering component of Transaction Services allows a third party application acting as a web service consumer to create checks within the Simphony system.

4.8.1 CalculateTotals

Given a set of check details, this web method calculates the various check totals.

4.8.2 PostNewTransaction

Given a set of check details, this web method creates a check in Simphony First Edition, sends menu items to order devices, authorizes credit card payments, and performs any other POS operations as dictated by the POS configuration.

The posted check is logged and is accessible through the PMC Check Journal Report.

4.8.3 PostToTransaction

Allow multiple service rounds for adding menu items and check detail items to existing open checks.

The posted check is logged and is accessible through the PMC Check Journal Report.

4.9 Error Codes

This list contains all possible errors that Transaction Service may return. Errors may return a user friendly message that tries to explain the problem, or a solution to correct the problem.

Error Code	Message
AppInitInProgress	
CCAuthDeclined	Credit card authorization declined.
CCAuthDeclinedWithMessage	Credit card authorization declined with a message '<message>'.
CCServerDown	Processing of the request has failed.
CheckEmployeeNumberMismatch	Check <check_number> {<check_guid>} returned with Employee Number <check_employee_number> which does not match search criteria Employee Number <search_employee_number>.

CheckNotFound	Check Not Found.
	Property=<property_object_number>
	RVC=<revenue_center_object_number>
	Workstation=<ts_workstaion_id>
	Check GUID=<search_check_guid>
	Closed Days=<search_checks_closed_for_n_days>
	Table=<search_table_number>
	Employee=<search_employee_number>
	Check Status=<check_close_status_code>

CheckOpenedOnSystem	
CheckTableNumberMismatch	Check <check_number> {<check_guid>} returned with Table Number <check_table_number> which does not match search criteria Table Number <search_table_number>.
ConnectionDown	Processing of the request has failed.
CreditCardPreambleNoMatchTender	Credit Card Number does not match any tender media preamble configuration.
DataOutOfRange	Data out of range. Data <data_name> holds up to <max_range>, found <exact_range>.
DetailDoesNotSupportTriggeredEvents	
DuplicateLineNumber	
EGatewayClientStartError	Start EGateway client failed. Reason:
EGatewayClientStopError	Stop EGateway client failed. Reason: <egw_error_code>

Error Code	Message
EGatewayConnectionError	EGateway connection Error. Reason: <egw_error_code>
EGatewayConnectionNotInPool	EGateway connection does not exist in the connection pool. URL: <egw_url>
EGatewayWaitConnectionTimeout	Wait for new connection timed out. URL: <egw_url> Timeout: <timeout> seconds
EmployeeClockIOStatusMismatch EmployeeIDMismatch EmployeeRVCMismatch	Employee <employee_last_name>, <employee_last_name>(<employee_number>) returned with <returned_employee_data> which does not match search criteria <searching_data>.

EmployeeNotFound	Employee not found.
	Property=<property_object_number>
	RVC=<revenue_center_object_number>
	WS=<ts_workstaion_id>
	ID=<employee_id>
	Employee=<employee_number>
	Clocked=<clocked_status>

FailedDataStoreInitialization	Application failed to initialize its local database.
FailedErrorTranslationInitial	Failed translation initialization. Reason: <error_info>
FailedPostCARequest	Failed to acquire CA authorization.
FailedInitialization	Application has failed to start.
FailedLoggerInitialization	Application has failed to initialize logger instance.
FailedSecurityAPIInitialization	Failed to initialize Security API. Error: <security_api_error>
InitialCCAuthNotFound	Initial credit card authorization not found. Account Last 4 Digits:<last_4_digits_of_initial_authorization_credit_card_number> Initial Credit Card Authorization Detail Line Number:<initial_credit_authorization_detail_line_number>
InvalidArguments	Invalid arguments. <procedure_name> Expect: <expected_arg_list> Actual: <actual_arg_list>
InvalidAutoServiceCharge	Invalid automatic service charge object number <invalid_service_charge_object_number> for RVC: <revenue_center_object_number> - <revenue_center_name>.
FailedSubmitPrintJob	Failed to submit print job. Error: <printing_controller_error_code> Guid: <print_job_guid>
InternalCommunicationError	
InternalProcessingError	Processing of the request has failed.
InvaildAuthCode	Processing of the request has failed.
InvalidCreditCardExpirationDate	An invalid credit card expiration date has been entered.
InvalidCreditCardHost	Processing of the request has failed.
InvalidCreditCardNumber	An invalid credit card number has been entered.
InvalidClientName	An invalid request has been made.
InvalidClosedDays	
InvalidConfigInfoRequestType	

Error Code	Message
InvalidConfigInfoType	An invalid request has been made.
InvalidCustomerInfo	
InvalidDeliveryTime	
InvalidDetailLine	An invalid request has been made.
InvalidDetailLineType	An invalid request has been made.
InvalidEGatewayConnectionURL	Invalid EGateway URL. URL: <config_url>
InvalidEmployeeNumber	
InvalidGuestCount	
InvalidLineNumber	
InvalidLogZone	Invalid Log Zone.
InvalidMenuItemPrice	
InvalidOrderTypeNumber	
InvalidPropertyNum	An invalid request has been made.
InvalidRvcNum	An invalid request has been made.
InvalidServingPeriod	
InvalidServiceChargeTaxClass	Invalid service charge tax class object number <invalid_tax_class_object_number>. Property: <property_object_number> - <property_name> RVC: <revenue_center_object_number> - <revenue_center_name> Service Charge: <service_charge_object_number> - <service_charge_name>
InvalidTableNumber	
InvalidTenderMedia	Invalid tender media object number <tender_media_object_number> for Property: <property_object_number> - <property_name>.
InvalidTenderMediaType	Invalid tender media type.
InvalidTranslationSpecifier	
MaximumAutofireChecksAllowed	
MenuItemNotActiveOnMenuLevel	Menu Item Definition is not active on Check Detail Item Menu Level. Object Number=<menu_item_definition_object_number> Sequence Number=<menu_item_definition_sequence_number> Definition Active Main Levels=<definition_active_main_levels> Definition Active Sub Levels=<definition_active_sub_levels> Detail Item Line Number=<check_detail_line_number> Detail Item Main Level=<check_detail_main_level> Detail Item Sub Level=<check_detail_sub_level>
MissingDetailLinesElement	
MissingTransactionElement	
MissingTransactionHeaderElement	
MissingTaxExemptReference	Tender detail item at position <detail_line_number>, with ID/object number <tender_media_object_number>, has tax exemption applied, and is missing required tax exemption reference information.
NoDataFound	No data returned.
NoRequestHeader	
NotImplemented	<procedure_name> not Implemented.
NullInput	
PaidPartially	
PostingDiscountDetailNotSupported	Discount detail posting not supported.

Error Code	Message
PreviousRoundItemFound	Previous round <detail_item_type>(<detail_id>) detail item found.
PriceMenuItemWithZeroAmount	
SecurityInitFailed	
ServiceTotalTenderMediaNotFound	No service total tender media configured for Property: <property_object_number> - <property_name>.
Success	Successful
TipServiceChargeNotConfig	Charged tip service charge not configured.
TransactionEmployeeNotFound	Transaction employee <employee_number> not Found.
TranslationFileNotAvailable	An invalid request has been made.
UnhandledException	An unexpected error has occurred.
UnknownCreditCardType	Unknown credit card type found in payment method.

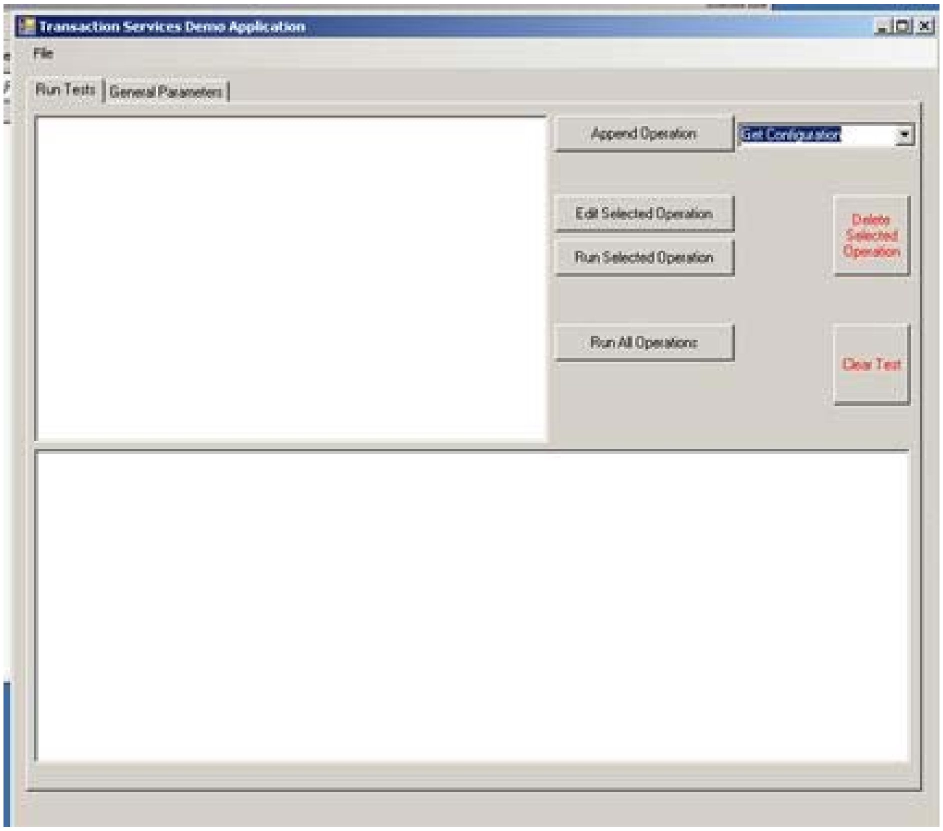
Demo Application

The demo application is a useful, but not necessary, part of the Transaction Services application. It is useful for demonstrations and testing purposes. The Transaction Services demo client can be run to call the web methods: CalculateTotals, GetConfigInfo, GetTransaction, PingServer, PostNewTransaction, and PostToTransaction. If the demo application does not function on a customer's property, it does not necessarily mean that a customer's application fails to post transactions to Simphony First Edition.

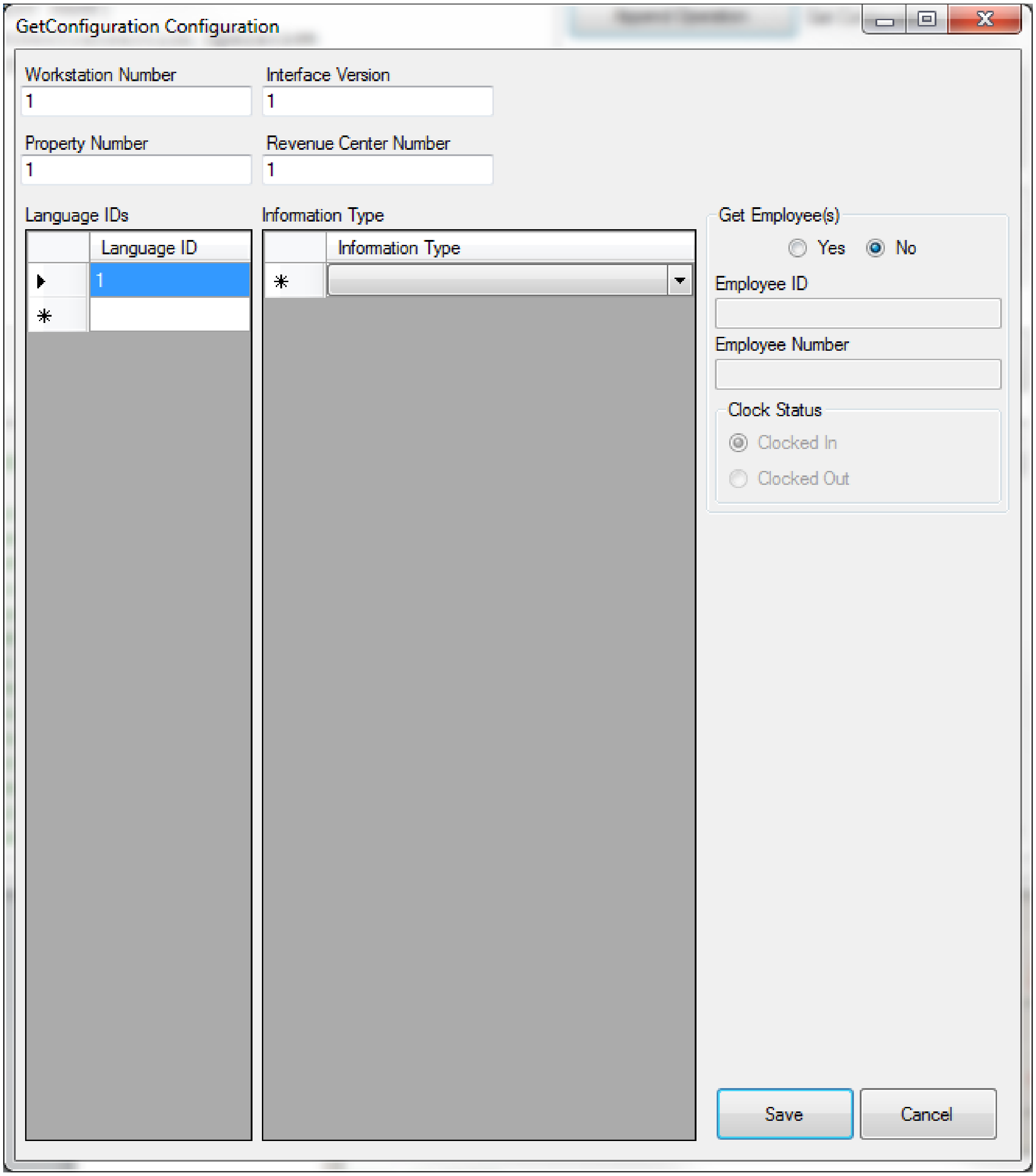
The Demo client application with source get gets installed in the directory: `C:\Inetpub\wwwroot\TSWebService\Sample Project`.

4.10 GetConfigInfo using Demo Client Application

1. Select **Get Configuration** from the drop-down list in the Transaction Service Demo Application.



2. On Get Configuration Window, change the revenue center **Number** field with the Revenue Center number and click **Save**.



GetConfiguration Configuration

Workstation Number: 1

Interface Version: 1

Property Number: 1

Revenue Center Number: 1

Language IDs

	Language ID
▶	1
*	

Information Type

	Information Type
*	

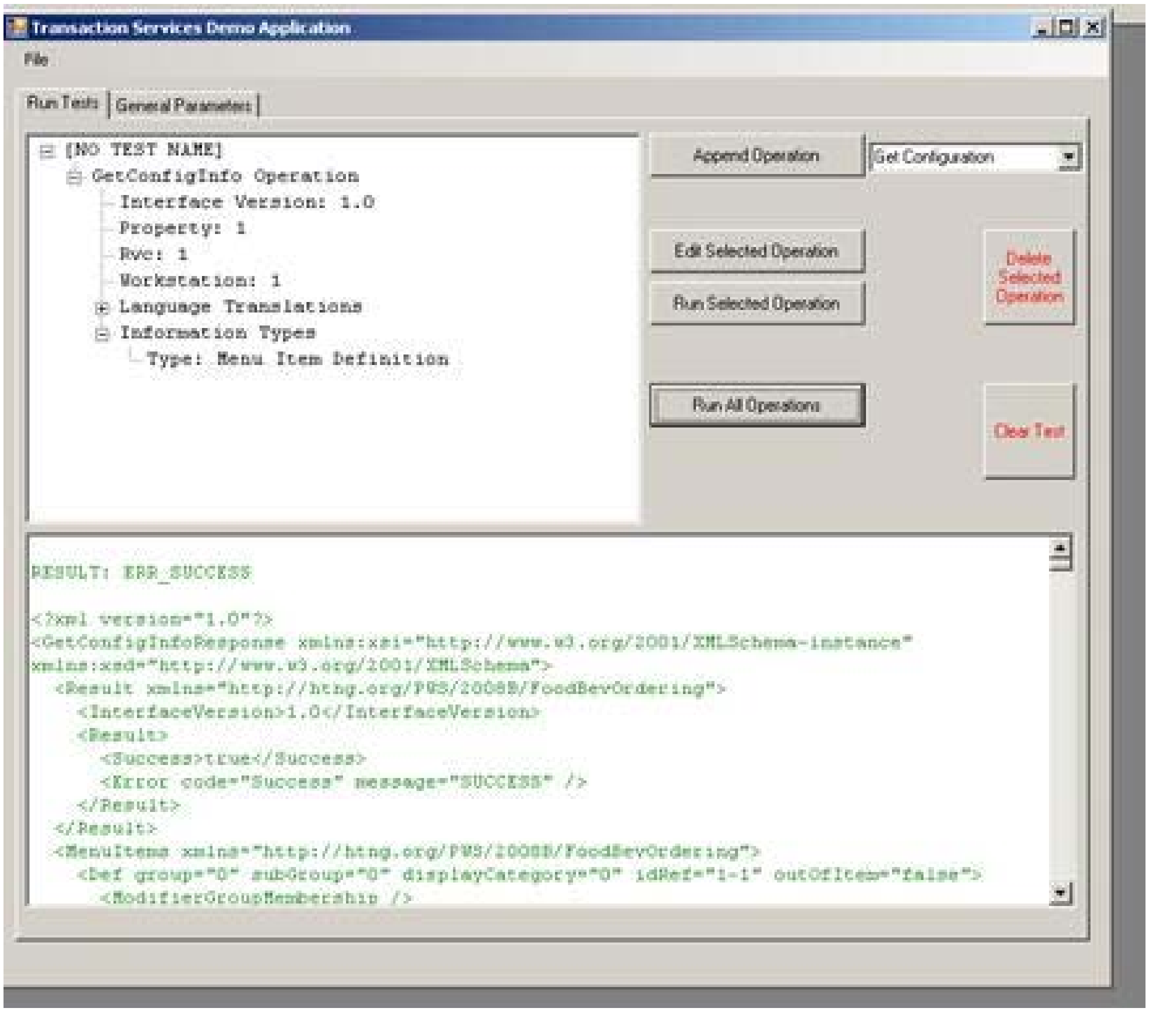
Get Employee(s): ○ Yes ● No

Employee ID:

Employee Number:

Clock Status: ● Clocked In ○ Clocked Out

3. On the Transaction Service Demo Application window, click **Run All Operations**.



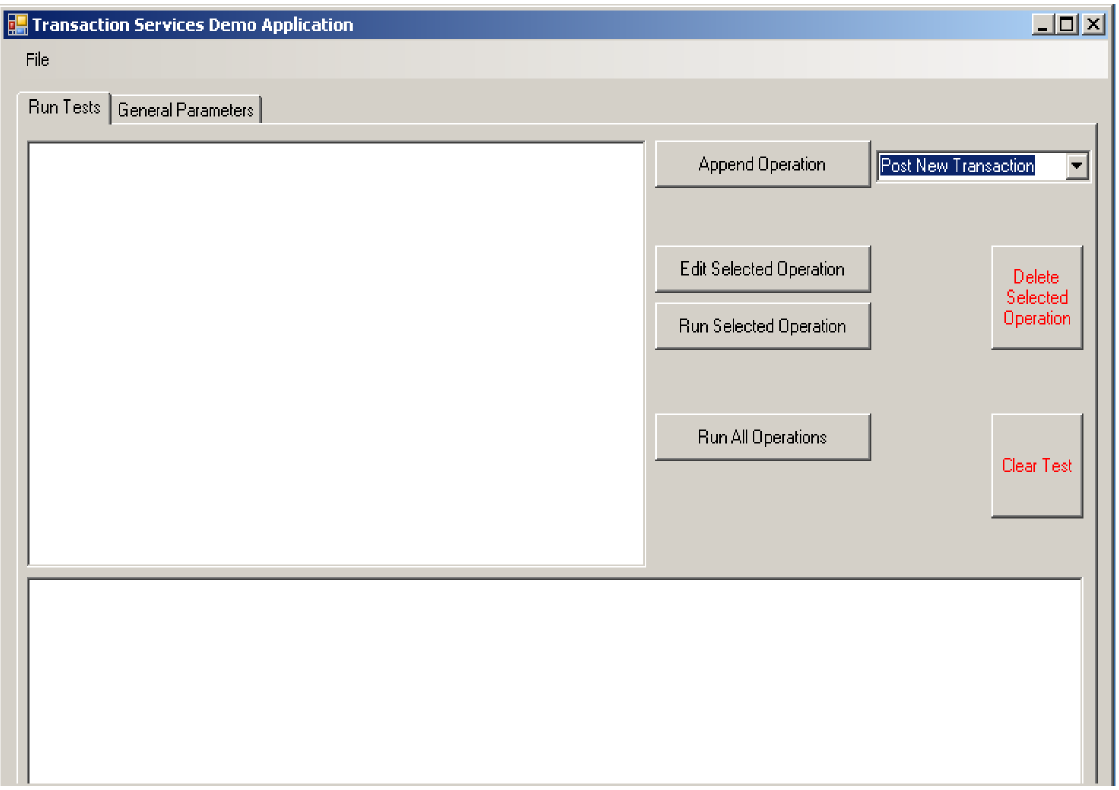
```
⊟ [NO TEST NAME]
   ⊟ GetConfigInfo Operation
      Interface Version: 1.0
      Property: 1
      Rvc: 1
      Workstation: 1
    ⊞ Language Translations
    ⊟ Information Types
        Type: Menu Item Definition
```

```
RESULT: ERR_SUCCESS

<?xml version="1.0"?>
<GetConfigInfoResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Result xmlns="http://htng.org/PWS/2008B/FoodBevOrdering">
    <InterfaceVersion>1.0</InterfaceVersion>
    <Result>
      <Success>true</Success>
      <Error code="Success" message="SUCCESS" />
    </Result>
  </Result>
  <MenuItems xmlns="http://htng.org/PWS/2008B/FoodBevOrdering">
    <Def group="0" subGroup="0" displayCategory="0" idRef="1-1" outOfItem="false">
      <ModifierGroupMembership />
```

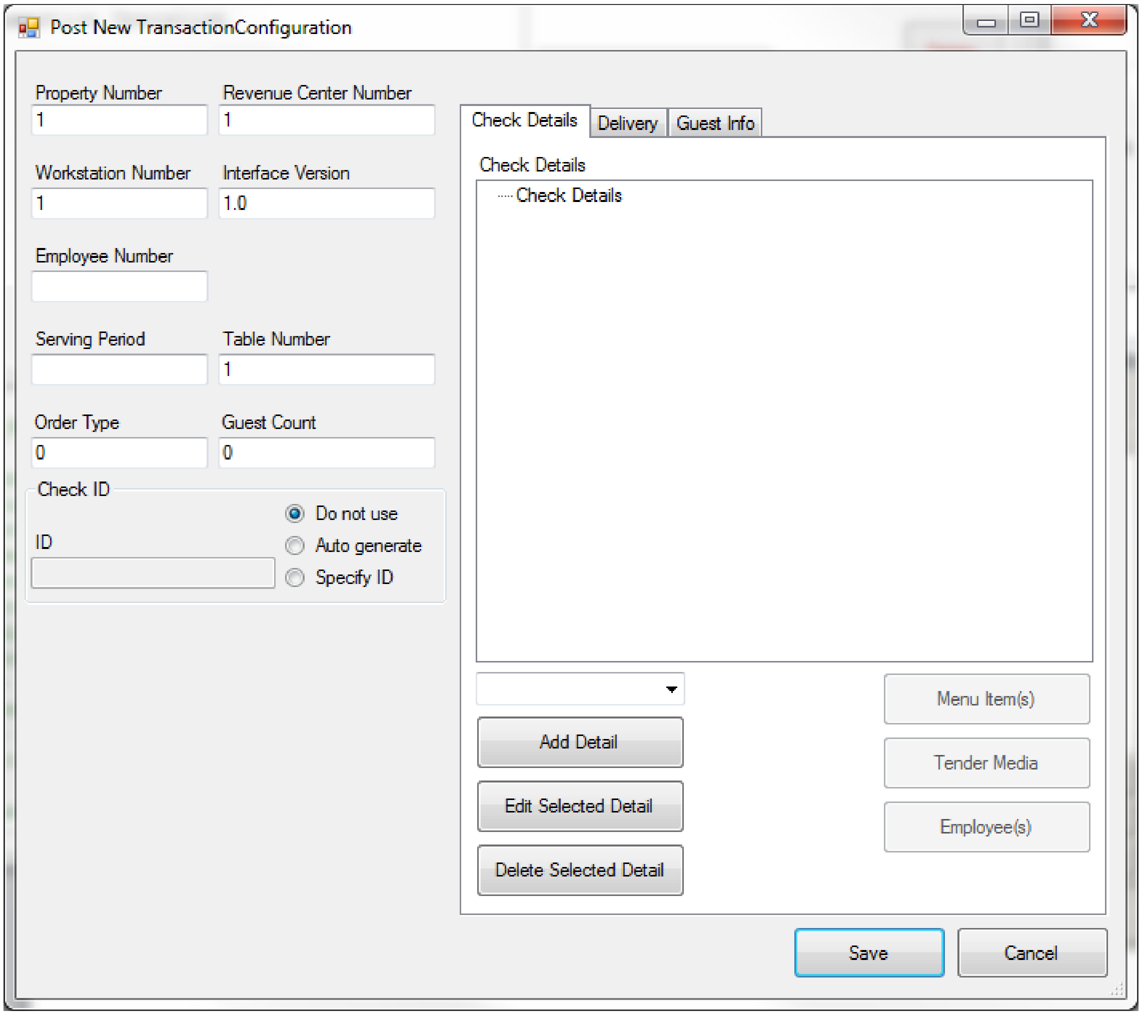
4.11 PostNewTransaction using Demo Client Application

1. Select **Post New Transaction** from the drop-down list in the Transaction Service Demo Application.



2. Click **Append Operation**. This opens the Post New Transaction Configuration window.

3. To add a menu item, select a Menu Item from the drop-down list in the Post New Transaction Configuration window, and then click **Add Detail**. This opens the Menu Item Edit Window.

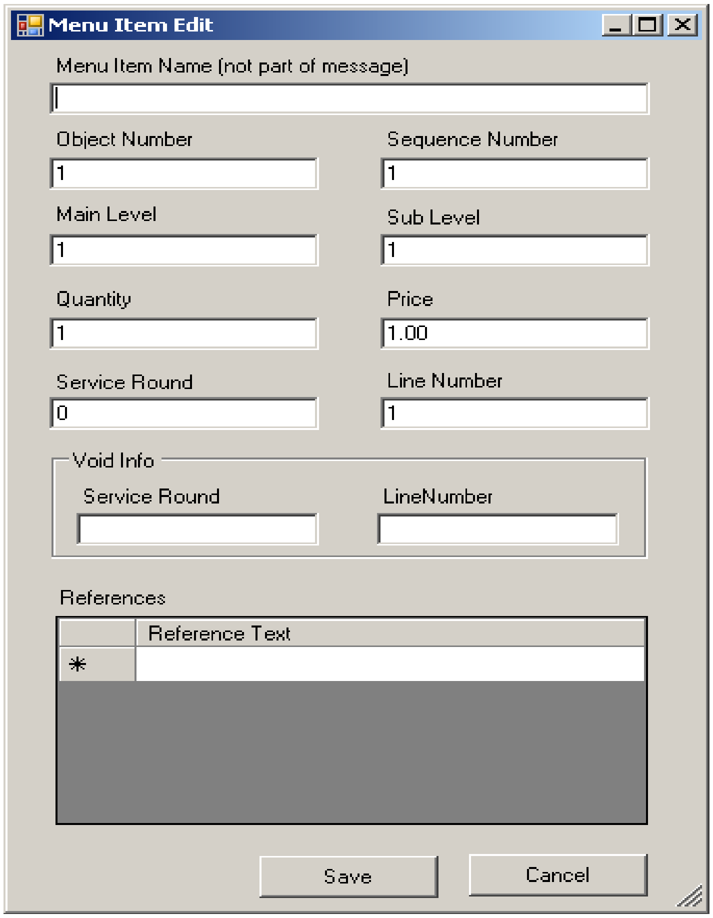


Post New TransactionConfiguration

Property Number: 1
Revenue Center Number: 1
Workstation Number: 1
Interface Version: 1.0
Employee Number:
Serving Period:
Table Number: 1
Order Type: 0
Guest Count: 0

Check ID
ID:
● Do not use
○ Auto generate
○ Specify ID

4. Enter the Object Number that matches a menu item in database. Click **Save** on the Menu Item Edit Window



Menu Item Edit

Menu Item Name (not part of message):

Object Number: 1	Sequence Number: 1
Main Level: 1	Sub Level: 1
Quantity: 1	Price: 1.00
Service Round: 0	Line Number: 1

Void Info

Service Round:

LineNumber:

References

	Reference Text
*	

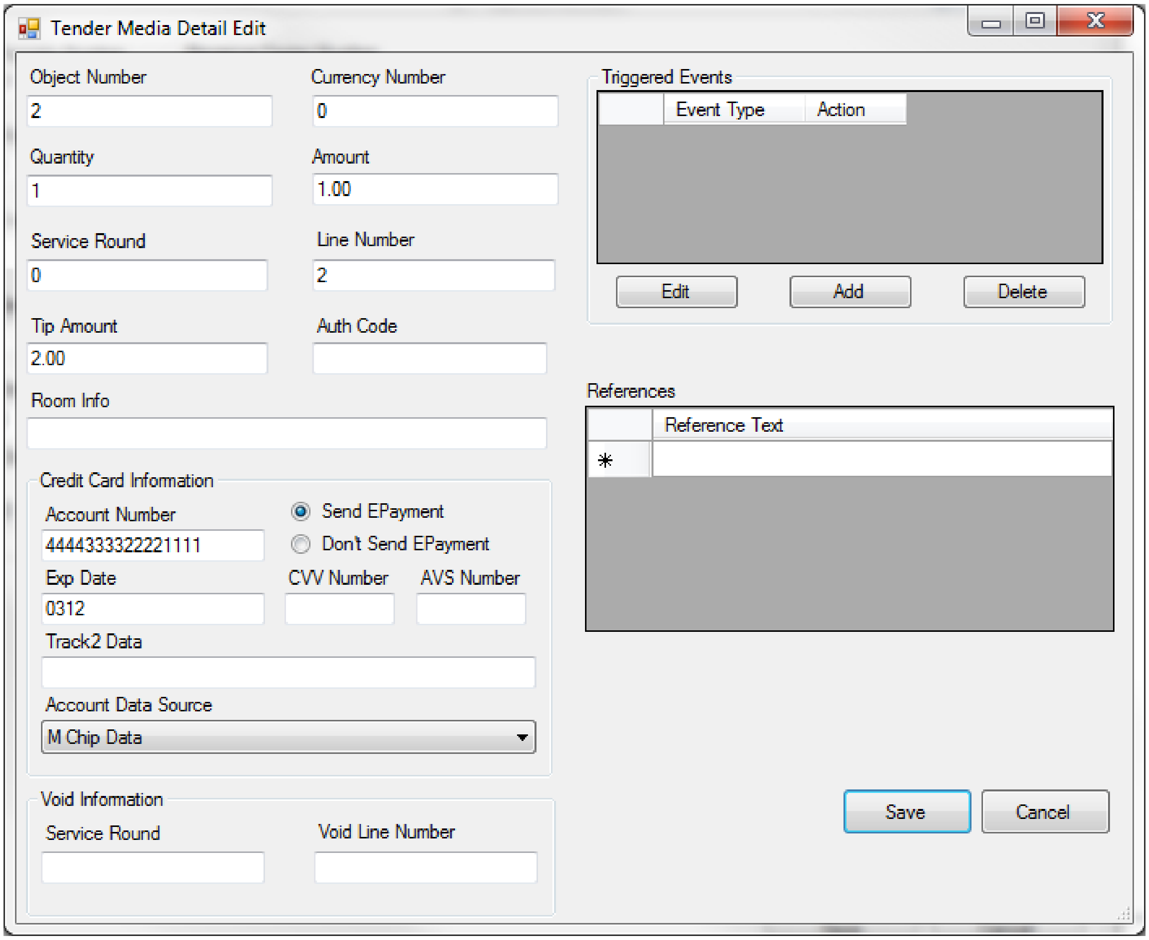
5. To add a Tender, select **Tender** from the drop-down list in the Post New Transaction Configuration window, and then click **Add Detail**.

   For Cash transactions:

   - Enter the Tender object number (2 in sample database)
   - Enter line number as 2
   - Enter the price
   - Select radio button **Don’t Send EPayment**.

   For Credit Card transactions:

   - Enter the Tender object number (3 in sample database)
   - Enter line number as 2
   - Enter the price
   - Select radio button **Send EPayment**.



Tender Media Detail Edit

Object Number: 2

Currency Number: 0

Quantity: 1

Amount: 1.00

Service Round: 0

Line Number: 2

Tip Amount: 2.00

Auth Code:

Room Info:

Credit Card Information

Account Number: 4444333322221111

● Send EPayment

○ Dont Send EPayment

Exp Date: 0312

CVV Number:

AVS Number:

Track2 Data:

Account Data Source: M Chip Data

Void Information

Service Round:

Void Line Number:

Triggered Events

	Event Type	Action

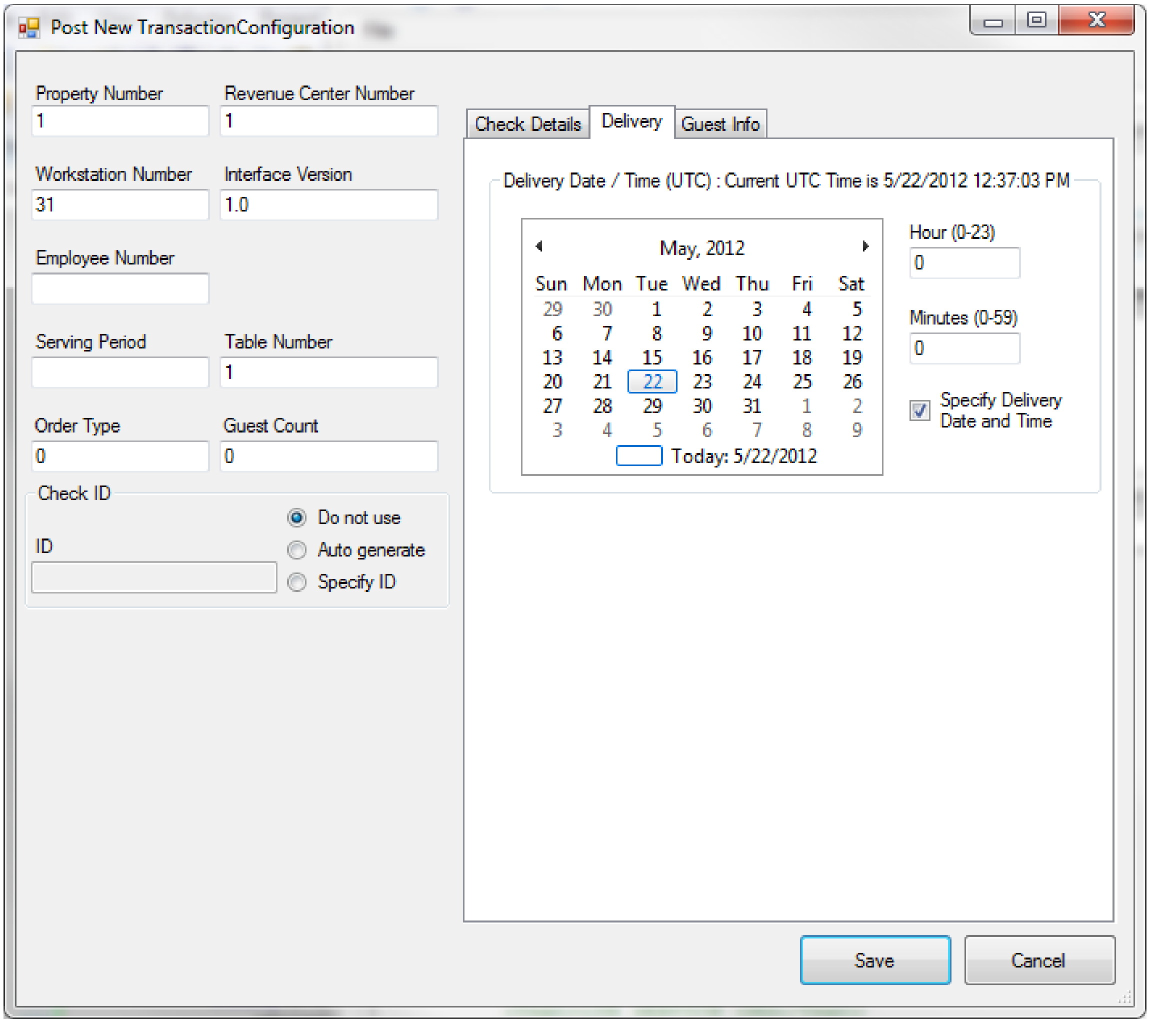
References

	Reference Text
*	

6. Click **Save** on Tender Media Detail Edit Window.

7. Click **Save** on Menu Item Edit Window.

8. For Autofire checks, defines the time to fire under the Delivery tab.



Property Number: 1

Revenue Center Number: 1

Workstation Number: 31

Interface Version: 1.0

Employee Number:

Serving Period:

Table Number: 1

Order Type: 0

Guest Count: 0

Check ID

ID:

● Do not use
○ Auto generate
○ Specify ID

Delivery Date / Time (UTC) : Current UTC Time is 5/22/2012 12:37:03 PM

May, 2012

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Today: 5/22/2012

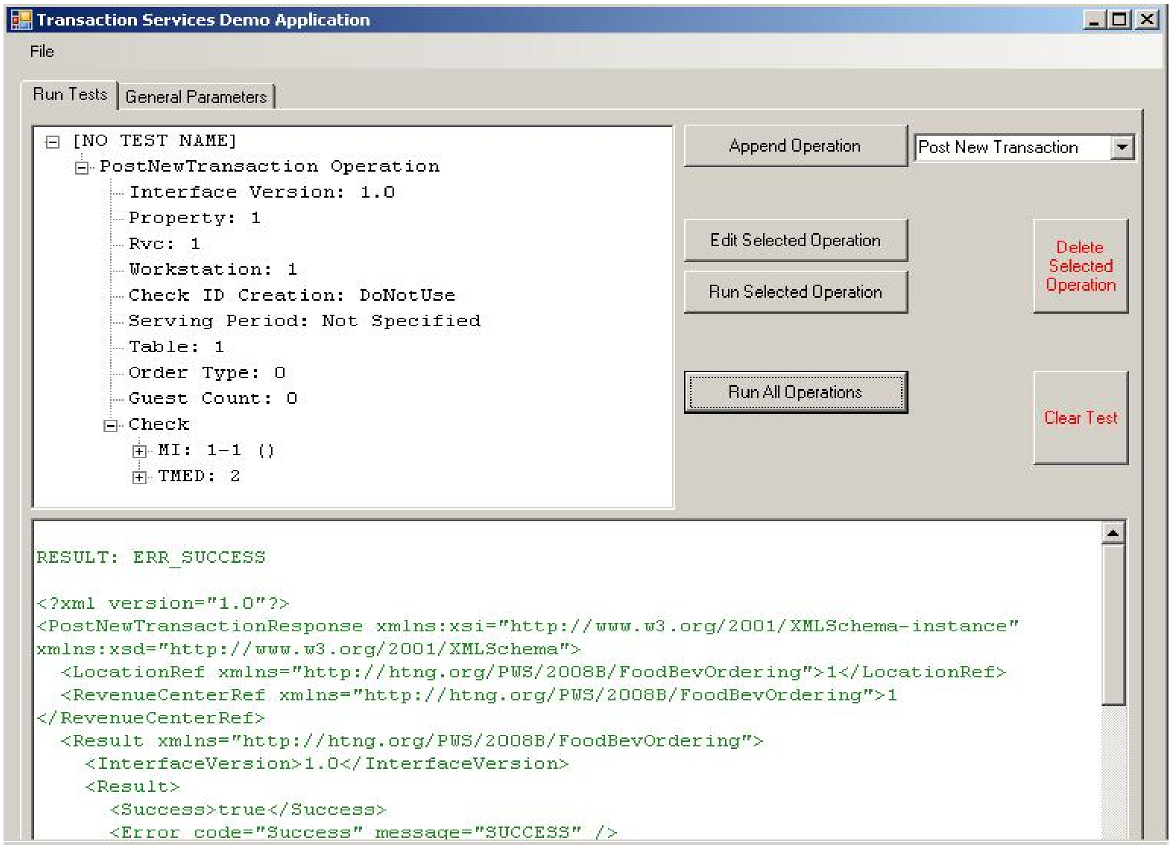
Hour (0-23): 0

Minutes (0-59): 0

☑ Specify Delivery Date and Time

9. Click **Save** on Post New Transaction Configuration window.

10. On Transaction Service Demo Application window, click **Run All Operations**.



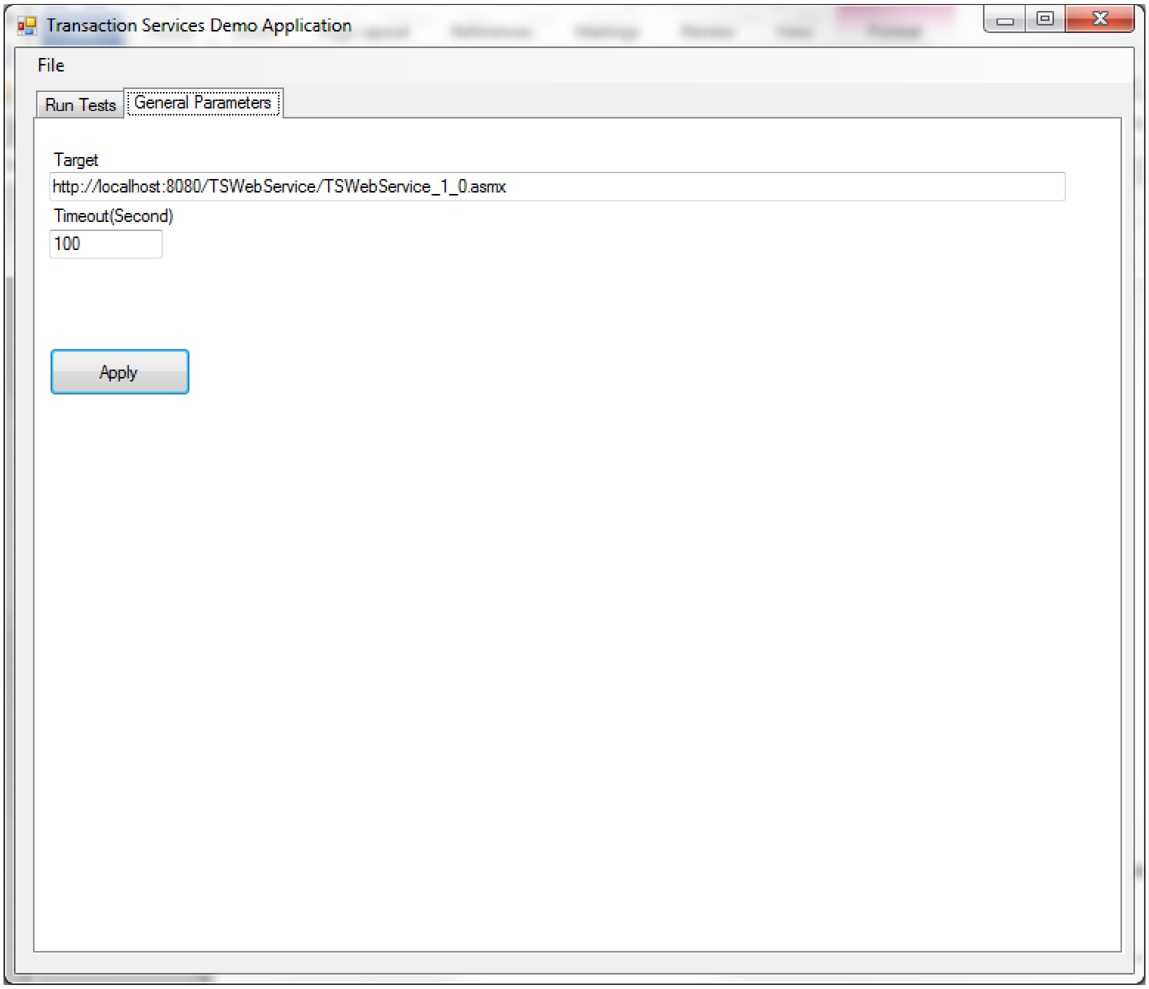
```
⊟ [NO TEST NAME]
   ⊟ PostNewTransaction Operation
       Interface Version: 1.0
       Property: 1
       Rvc: 1
       Workstation: 1
       Check ID Creation: DoNotUse
       Serving Period: Not Specified
       Table: 1
       Order Type: 0
       Guest Count: 0
     ⊟ Check
         ⊞ MI: 1-1 ()
         ⊞ TMED: 2
```

```
RESULT: ERR_SUCCESS

<?xml version="1.0"?>
<PostNewTransactionResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <LocationRef xmlns="http://htng.org/PWS/2008B/FoodBevOrdering">1</LocationRef>
  <RevenueCenterRef xmlns="http://htng.org/PWS/2008B/FoodBevOrdering">1
</RevenueCenterRef>
  <Result xmlns="http://htng.org/PWS/2008B/FoodBevOrdering">
    <InterfaceVersion>1.0</InterfaceVersion>
    <Result>
      <Success>true</Success>
      <Error code="Success" message="SUCCESS" />
```

4.12 Post Requests to different Transaction Services

It is possible to post requests to multiple Transaction Services which have different URLs. In order to do that the target URL has to be updated under General Parameters tab. By changing this target URL, Demo Client will then able to post requests to a different Transaction Service.



Target

http://localhost:8080/TSWebService/TSWebService_1_0.asmx

Timeout(Second)

100

Limitations

4.13 Limited Check Operations

Transaction Services cannot pickup checks, adjust closed checks, reopen checks or split checks. Essentially, once Transaction Services has open and closed a check, Transaction Services no longer has access to the check.

4.14 No Support for PMS

Transaction Services does not support room charges or PMS-like payments.

4.15 No Support for Stored Value cards

Transaction Services does not support stored value card payments.

4.16 No offline mode support

Transaction services do not operate while disconnected from the data center. Though workstation clients and other components on property operate while offline from the datacenter, Transaction Services does not do so.

4.17 Limited to single workstation and single revenue center

If utilizing Simphony First Edition prior to the 1.7 release, Transaction Services does not handle requests that have different workstation ID or revenue center ID information other than the configured workstation along with the revenue center assigned to the workstation. Transaction Services only downloads data belonging to the revenue center of the configured workstation. A request that belongs to another workstation or revenue center, results in an error that explains that the data was not found.

With the release of Simphony First Edition versions 1.7.x and later:

- Simphony First Edition supports configuring a Transaction Services (POS API) workstation to be assigned to up to eight revenue centers.
- Transaction Services is supported running on the same hardware devices as Simphony First Edition point of sales (POS) clients that also run SAROps.

4.18 No multiple workstations or multiple revenue centers

Transaction services does not support multiple workstations settings at the same time, or multiple revenue centers. However, it is possible to configure another instance of Transaction Service on the same computer to use another workstation and assign a different revenue center.

5 Troubleshooting

5.1 Logging

5.1.1 Location of Logs

Transaction Services is typically installed in a location like “`C:\Logs`”. Transaction Services logs the incoming and outgoing SOAP messages. This is very useful in determine what the WSCs are sending and receiving.

5.2 Common Problems and their Solutions

5.2.1 Application failed to initialize its local database.

This error message indicates that Transaction service was not able to download the local datastore. The reasons for this error message could be Transaction Service could not communicate with EGateway server or frequent network disruption. Most of the times, this is resolved by an iisreset on the Transaction service machine. If iisreset does not fix this problem, check the network settings and make sure that Transaction service can communicate with EGateway server and wait for the DefaultDBRefreshInterval time set in web.config file.

5.2.2 Error: Encryption Key Not Found.

The customer has not run TSConfig.exe or deleted the token id field is missing in web.config file. To resolve this issue, re-run TSConfig.exe.

5.2.3 Error: Failed to encrypt credit card data.

This error method is returned when Transaction service fails to encrypt the credit card data. This occurs in the early processing of credit card. The main reason for this is REGREAD_ERROR, which means the Network service user or ASP.NET user has no permission to registry keys userdata and/or userdataXXX (XXX being workstation id). To resolve this, re-run TSConfig.exe

5.2.4 Error: Authorization Failed.

This error message is returned in a Credit card transaction when Transaction service was able to communicate with Credit card service host but did not receive an authorization code. Check if the system running as credit card service host can communicate with the credit card processing company.

5.2.5 Common error messages due to incorrect Requests

Some of the common error messages returned due to incorrect request object received are:

- Invalid 'LineNumber' in check detail
  Line numbers should be unique and incremented for each detail item

- Failed to create check. Check was not paid in full
  The check was not paid in full. Transaction service cannot leave the check open and the checks need to be paid in full when opened.

- Could not create MenuItemDetail from Object.
  The menu item could not be found for the menu item object number sent to Transaction service.

- Invalid 'PriceLevelIDRef' in menu item detail
  The price level are not defined in EMC for this menu item.

5.2.6 Why don't I see the WSDL file when I request WSDL

For a web service, any WSC can request a WSDL through the browser by browsing to *http://localhost:8080/TSWebservice/TSWebservice_1_0.asmx?WSDL*. But this version of Transaction service does not returned a well formed WSDL as a normal web service would return because the software was built by reverse engineering using WSDL.exe command to generate the server side stub/code using the existing WSDL file. WSC can refer to the SOAP xml messages by clicking on each web method or refer to *Configure Transaction Service.doc*