Oracle® Hospitality RES 3700

POS Transaction Services

Release 5.7.0200.4923
F13800-04
June 2021

Oracle Hospitality RES 3700 POS Transaction Services, Release 5.7.0200.4923

F13800-04

Contents

Preface

Purpose

This document is intended for 3rd Party Developers who are implementing specialized POS Client Applications utilizing the Oracle Hospitality RES 3700 POS API.

The interface will allow access to RES 3700 POS Transaction Services enabling 3rd Party developers to create specialized POS clients. The API allows the specialized POS clients to access POS functions such as opening tables/guest checks, starting orders and applying payments and discounts.

Some examples of specialized POS clients are:

- Remote or local guest ordering from Kiosks
- Remote guest ordering or centralized order dispatch utilizing Web Services
- Open table/guest check control and guest experience time management via Table Management Systems
- Guest payment approval using mobile phones or PDAs

Audience

This document is intended for:

- Third party developers
- Database and System administrators

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

https://support.oracle.com

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received and any associated log files
- Screenshots of each step you take

Documentation

Product documentation is available on the Oracle Help Center at http://docs.oracle.com/en/industries/food-beverage/

Revision History

Date	Description of Change
January 2019	• Initial publication. • Updated the *POS API Employee* and *Enabling Logging* sections in Chapter 1. • Added the *Security Considerations* section to Chapter 1.
March 2019	• Updated *Enabling Logging* chapter to reflect new steps.
June 2021	• Updated the *ResPosAPI_GuestCheck Struct Reference* section in Chapter 3 by adding long check status bits.

1 Installation

Transaction Services is installed with RES. See *RES 3000 Platform Requirements* for requirements for this release.

Licensing Configuration

The Transaction Services feature is licensed, and requires activation via the License Manager Application.

Activating the License:

1. Access License Manager from MICROS Applications | Utilities Menu.
2. Locate the appropriate Transaction Services features.
3. Enter the check box to enable the license.
4. Click “Reload” button.

POS API Employee

You will need to decide which employee(s) should be used when submitting orders to the API. Any valid employee may be used. You may decide to define one or more employees in the RES 3700 application as a “POS API Employee”. Originally, this Employee Class option was intended for security and reporting purposes. This option is obsolete now, as any employee id can be used when submitting orders to the API.

Define Transaction Services Employee Class

1. Access POS Configurator from the MICROS Applications Menu.
2. From the Employee | Employee Classes form, insert a new employee class.
3. On the Employee | Employee Classes | Transaction | Transaction Control form, enable the option for POS API Employee.
4. Assign the new employee class to the appropriate employees.

> **WARNING:**
>
> When enabling this option, most of the other options of an Employee Class become disabled, but their configured state still takes effect.

Enabling Logging

Transaction Services posts errors and various status messages when logging is enabled. The following parameters control the logging behavior. These are registry entries on the RES 3700 application server.

Registry key

HKLM\Software\[WOW6423Node\]MICROS\3700\POSAPI

Registry values

DWORD Verbosity

0 (default) is the min, 10 is the max. At higher values, transaction services logs increasingly more information.

DWORD LocalLogging

0 – (default) logging is directed to the 3700d.log
1 – logging is directed to .\etc\ResPosApi.log

The verbosity may also be set using MICROS Control Panel.

1. From the View menu, enable the View Verbosity option.
2. Select the Server in the tree view on the left.
3. Select the Verbosity tab on the right.
4. Select **ResPosApi** from the list of modules.
5. Use the slider at the bottom to alter the verbosity.

Security Considerations

The web services interface exposed by RES 3700 Transactions Services does not provide a means of encrypted communication. It is expected that this interface will only be exposed on a closed, internal network. The Oracle Hospitality RES 3700 Security Guide contains more information.

2 Components and Services

Transaction Services provides core POS functions such as:

- Opening tables and guest checks
- Adding detail items (menu items, discounts and payments) to guest checks
- Posting guest check detail to the database

Transaction Services are accessible on a local client PC using a COM interface or on a remote client using a web service.

Local COM Interface

Calling Conventions

There are two types of parameters passed to the API: [in] and [in, out] parameters. All parameters are required! However, if you do not wish to use one of the parameters, simply create the structure and set all of its members to zero.

An example of the C++ code to define a discount structure would be as follows:

```
ResPosAPI_Discount testDiscount;
ZeroMemory( &testDiscount, sizeof testDiscount);
```

If a check does not contain a subtotal discount, then you would pass the address of this structure to the API - everything will be zero. To add a discount, fill in the appropriate members of the discount object.

Passing NULL is Discouraged

In some programming languages, NULL is supported for unused parameters. Please avoid using NULL for this purpose. In future releases, the API may become more stringent on NULL parameters which may break your code! As a rule, it's better to simply pass a zero-initialized object than to use NULL.

Web Services

Transaction Services has also been packaged as a web service allowing remote client's transaction access to the POS Server. The web service exposes the same core methods supported by the local COM interface. For ease of implementation, all of the parameter names are the same between the COM interface and the web service.

Two additional methods are supported in the web service:

- UserExecuteAssembly

  Allows remote client to execute a .Net assembly on the POS Server

- UserExecuteDll

Allows the remote client to execute a c style dll on the POS Server

Service Description

Once the API has been installed, the formal service description is available by accessing the following URL:

http://localhost/ResPosApiWeb/ResPosApiWeb.asmx

This should show the .NET default Welcome Screen for the ResPosApiWebService.

Printing Services

The API supports printing to remote and local order devices. When a check is opened thru the API and posted to the RES database, the menu items will print on the remote order and local devices based on the default Workstation definition and the menu item print class. There should be no difference between how an API check prints versus a check opened directly by the user on the POS order devices. Local Guest check printing is not supported at this time thru the API.

Condiment Handling

Transaction Services is significantly less strict with condiments than POS Operations. The rules and the results are fairly simple.

Rules

1. Anything specified as a condiment must be defined as a condiment in the RES database.

   The API and POS Operations need to be able to work side by side so condiments must be defined in the RES database, otherwise the neither the API or POS Operations will understand what to do with the menu item.

2. Any condiment can be placed on any parent.

   The API provides the ultimate flexibility to describe orders. This is one of the most powerful features the API provides. The API will make your order compatible with the in store POS Operations.

3. All condiments will be marked in the API as a required condiment.

   This is done so that sorting of allowed condiments at the POS Operations terminal will not re-order your API condiment selections.

Glossary of Terms

If you encounter any terminology that you would like to see listed in the Glossary, let us know.

Table 2-1 - Glossary

Term	Description
Object Number	User-defined value that uniquely identifies a definition (menu, revenue center, etc.) item in the MICROS RES database. The range for object numbers is 1-9999999.
Sequence Number	System-defined value that can uniquely identify a definition item (menu, revenue center, etc.) in the MICROS RES database. The range for sequence numbers is 1-9999999.

3 Class Documentation

IGetCheckInfo Interface Reference

Check Information Interface.
Inherits IUnknown.
Inherited by CTransactionServices

Public Member Functions

- HRESULT GetOpenChecks ([in] long EmployeeObjectNum, [in, out]SAFEARRAY(ResPosAPI_CheckSummary)*ppCheckSummaryArray)

  Get Open checks on the point of sale system.
- HRESULT GetOpenChecksWithSeats ([in] long EmployeeObjectNum, [in, out]SAFEARRAY(ResPosAPI_CheckSummaryWithSeats)*ppCheckSummaryArrayWithSeats)

  Get Open checks with seats on the point of sale system.
- HRESULT GetPrintedCheck ([in] long CheckSeq, [in] long EmplObjectNum, [in] long TmedObjectNum, [in, out]SAFEARRAY(BSTR)*ppCheckPrintLines, [in, out] ResPosAPI_GuestCheck *pGuestCheck, [in, out] ResPosAPI_TotalsResponse *pTotalsResponse)

  Get a printed version of the Open Guest Check.
- HRESULT GetCheckSummaryWithSeats ([in] long CheckSeq, [in] BSTR EmplIdNum, [in, out] long *EmplObjectNum, [in, out] ResPosAPI_CheckSummaryWithSeats *pCheckSummary)

  Get a check summary with seats on the point of sale system.
- HRESULT GetGiftCardItemsOnCheck ([in] long CheckSeq, [in] long EmplObjectNum, [in]SAFEARRAY(ResPosApi_GiftCardItemDef) pRequestedGiftCardItems, [in, out]SAFEARRAY(ResPosAPI_GiftCardItem)*ppItems)

  Get gift card items on a check.
- HRESULT GetPaymentBySeat ([in] ResPosAPI_GuestCheck *pGuestCheck, [in] VARIANT_BOOL bGetVoidedPayments, [in, out] SAFEARRAY(ResPosAPI_SeatPayment)*pCheckSeatPayment)

  Get Payments on Check by Seat.

Detailed Description

Check Information Interface
This is the interface used for general check information.

Member Function Documentation

HRESULT GetOpenChecks ([in] long *EmployeeObjectNum*, [in, out] SAFEARRAY(ResPosAPI_CheckSummary)* *ppCheckSummaryArray*)

Get Open checks on the point of sale system

When using the GetOpenChecks, the employee object number can be specified to filter open checks by the employee. If zero is passed in the employee object field, then all open checks will be returned.

Parameters:

EmployeeObjectNum	**Zero for all employees or the specified employee**
ppCheckSummaryArray	Array of check summaries returned.

See Also:

ResPosApiTypes: ResPosAPI_CheckSummary Struct Reference

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT GetOpenChecksWithSeats ([in] long *EmployeeObjectNum*, [in, out] SAFEARRAY(ResPosAPI_CheckSummaryWithSeats)* *ppCheckSummaryArrayWithSeats*)

Get Open checks with seats on the point of sale system

When using the GetOpenChecksWithSeats, the employee object number can be specified to filter open checks by the employee. If zero is passed in the employee object field, then all open checks will be returned.

Parameters:

EmployeeObjectNum	**Zero for all employees or the specified employee**
ppCheckSummaryArrayWithSeats-	Array of check and seats summaries returned.

See Also:

ResPosApiTypes: ResPosAPI_CheckSummaryWithSeats Struct Reference

NOTE:

If an error occurs when picking up the check the ppCheckSummaryArrayWithSeats will be empty. This will be indication that the check information is not valid. There should always be at least one seat present if the check reading was successful.

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT GetPrintedCheck ([in] long *CheckSeq*, [in] long *EmplObjectNum*, [in] long *TmedObjectNum*, [in, out] SAFEARRAY(BSTR)* *ppCheckPrintLines*, [in, out] ResPosAPI_GuestCheck * *pGuestCheck*, [in, out] ResPosAPI_TotalsResponse * *pTotalsResponse*)

Get a printed version of the Open Guest Check

The guest check printer does not have to be configured for this method to return a guest check. The default format is 32 columns. If a 40 column print job is required the API workstation can be configured to print a guest check to a hard disk in 40 columns. The POS API will return the strings sent to the configured printer.

Tender media is required because the tender has several printing options that assist in formatting of the final guest check. See *Pos Configurator Tender Media* for details.

Formatting:
Several Format characters are left in the strings returned.

LINEFEED
0x17 is a linefeed character, the lead nibble will have the number of linefeeds desired sample code:

```
numLineFeeds &= 0x7F;
 wchar_t multiFeed[3] = {(wchar_t)0x17 ,numLineFeeds,0};
```

BOLD

0x11 Is the BOLD ON Character it is followed by a BOLD OFF character.

0x12 Is the BOLD OFF Character it is preceded by a BOLD ON character.

0x11 **Some Bold Text** 0x12

RED

0x13 Is the RED ON Character it is followed by a RED OFF character.

0x15 Is the RED OFF Character it is preceded by a RED ON character.

Red print is not often used with thermal printers.

Double Tilde ~~

The double tilde is followed by the name of a file for bitmap printing. This file is located in `\Micros\res\pos\bitmaps`

The ResPosAPI_GuestCheck structure is also passed but is only an out parameter. This way the developer will have all of the needed information about the check.

The ResPosAPI_TotalsResponse structure is also passed an out parameter to enable access to the check total information without having to parse the printed check lines.

Parameters:

CheckSeq	Specify the check sequence number desired.
EmplObjectNum	Specify the check or transaction employee that wishes to print the check.
TmedObjectNum	Specify the tender media to print the check with. A service total type is the only permitted type of tender media.
ppCheckPrintLines	Array of Printed Guest Check Lines. This may be in 32 or 40 column format.
pGuestCheck	The check structure to be returned.
pTotalsResponse	The totals structure to be returned.

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT GetCheckSummaryWithSeats ([in] long *CheckSeq*, [in] BSTR *EmplIdNum*, [in, out] long * *EmplObjectNum*, [in, out] ResPosAPI_CheckSummaryWithSeats * *pCheckSummary*)

Get a check summary with seats on the point of sale system

When using the GetCheckSummaryWithSeats the system will try to pick up and read the check with the given CheckSeq. It will first try the check as open. If this fails, it will try it as closed.

Parameters:

CheckSeq	
EmplIdNum	Pass in employee id (Password Id field in POS Cfg) and the matching EmplObjectNum will be returned in EmplObjectNum. Or pass in payroll id preceded with # sign. (i.e., #100100, where 100100 is the payroll_id).
EmplObjectNum	Pass in employee object number
pCheckSummary	The check summary with seats

See Also:

ResPosApiTypes: ResPosAPI_CheckSummaryWithSeats Struct Reference

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT GetGiftCardItemsOnCheck ([in] long *CheckSeq*, [in] long *EmplObjectNum*, [in] SAFEARRAY(ResPosApi_GiftCardItemDef) *pRequestedGiftCardItems*, [in, out] SAFEARRAY(ResPosAPI_GiftCardItem)* *ppItems*)

Get gift card items on a check

When using the GetGiftCardItemsOnCheck the system will try to pick up and read the check with the given CheckSeq. If successful, it will return a list of requested gift card items on the check.

Parameters:

CheckSeq	
EmployeeObjectNum	
requestedGiftCardItems	A list with gift card items to fetch
ppItems	A result set of gift card items

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT GetPaymentBySeat ([in] ResPosAPI_GuestCheck * *pGuestCheck*, [in] VARIANT_BOOL *bGetVoidedPayments*, [in, out] SAFEARRAY(ResPosAPI_SeatPayment)* *pCheckSeatPayment*)

Get Payments on Check by Seat

The ResPosAPI_SeatPayment structure is passed as out parameter to enable access to the check seat payment information.

Parameters:

pGuestCheck	**The check for which the information is requested**
bGetVoidedPayments	Flag indicating a request for voided payments
pCheckSeatPayment-	Array of payments by seat present on the check

See Also:

- ResPosApiTypes: ResPosAPI_GuestCheck Struct Reference
- ResPosApiTypes: ResPosAPI_SeatPayment Struct Reference

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

IPrintServices Interface Reference

IPrintServices is the main printing interface for RES 3700 Transaction Services
Inherits IUnknown.
Inherited by CPrintServices.

Public Member Functions

- HRESULT SubmitPrintJob ([in, out] ResPrintAPI_PrintJob *pPrintJob)

  Accepts a print job for printing.
- HRESULT FormatDoubleWide ([in, out] BSTR *pString)

  Formats desired string double wide.
- HRESULT FormatLineFeed ([in, out] BSTR *pString, [in] SHORT numLineFeeds)

  Formats a print job line feeds.
- HRESULT CheckPrintStatus ([in] long StatusID, [in, out] ResPrintAPI_PrintJobStatus *pStatus)

  CheckPrintStatus Checks the status of a print job.

Detailed Description

IPrintServices is the main printing interface for Transaction Services.

Member Function Documentation

HRESULT CheckPrintStatus ([in] long *StatusID*, [in, out] ResPrintAPI_PrintJobStatus * *pStatus*)

CheckPrintStatus checks the status of a print job

Remarks:
It is highly suggested to understand the ResPrintAPI_PrintJobStatus structure. Careful attention must be paid to this structure to get accurate information about the print job and system printer status.

ITransactionServices Interface Reference

The Main API interface.
Inherits IUnknown.
Inherited by CTransactionServices

Public Member Functions

- HRESULT CalculateTransactionTotals ([in, out] SAFEARRAY(ResPosAPI_MenuItem)*ppMenuItems, [in, out] SAFEARRAY(ResPosAPI_ComboMeal)*ppComboMeals, [in, out] ResPosAPI_SvcCharge *pServiceChg, [in, out] ResPosAPI_Discount *pSubTotalDiscount, [in] LONG revenueCenterObjectNum, [in] SHORT orderType, [in] LONG employeeObjectNum, [in, out] ResPosAPI_TotalsResponse *pTotalsResponse)

  Calculate transaction totals.

- HRESULT PostTransaction ([in, out] ResPosAPI_GuestCheck *pGuestCheck, [in, out] SAFEARRAY(ResPosAPI_MenuItem)*ppMenuItems, [in, out] SAFEARRAY(ResPosAPI_ComboMeal)*ppComboMeals, [in, out] ResPosAPI_SvcCharge *pServiceChg, [in, out] ResPosAPI_Discount *pSubTotalDiscount, [in, out] ResPosAPI_TmedDetailItem *pTmedDetail, [in, out] ResPosAPI_TotalsResponse *pTotalsResponse)

  Post transaction to the database.

- HRESULT AddToExistingCheck ([in, out] ResPosAPI_GuestCheck *pGuestCheck, [in, out] SAFEARRAY(ResPosAPI_MenuItem)*ppMenuItems, [in, out] SAFEARRAY(ResPosAPI_ComboMeal)*ppComboMeals, [in, out] ResPosAPI_SvcCharge *pServiceChg, [in, out] ResPosAPI_Discount *pSubTotalDiscount, [in, out] ResPosAPI_TmedDetailItem *pTmedDetail, [in, out] ResPosAPI_TotalsResponse *pTotalsResponse)

  Add or change transaction.

- HRESULT VoidTransaction ([in, out] ResPosAPI_GuestCheck *pGuestCheck)

  Void all of the items on the guest check

- HRESULT CheckPrintJobStatus ([in, out] SAFEARRAY(LONG)*ppJobIdList, [in, out] SAFEARRAY(ResPrintAPI_PrintJobStatus)*ppJobStatusList)

  Returns the status of a print job.

- HRESULT PostTransactionEx ([in, out] ResPosAPI_GuestCheck *pGuestCheck, [in, out] SAFEARRAY(ResPosAPI_MenuItem)*ppMenuItems, [in, out] SAFEARRAY(ResPosAPI_ComboMeal)*ppComboMeals, [in, out] ResPosAPI_SvcCharge *pServiceChg, [in, out] ResPosAPI_Discount *pSubTotalDiscount, [in, out] ResPosAPI_TmedDetailItemEx *pTmedDetailEx, [in, out] ResPosAPI_TotalsResponse *pTotalsResponse, [in, out] SAFEARRAY(BSTR)*ppCheckPrintLines, [in, out] SAFEARRAY(BSTR)*ppVoucherOutput)

  Post transaction to the database.

- HRESULT AddToExistingCheckEx ([in, out] ResPosAPI_GuestCheck *pGuestCheck, [in, out] SAFEARRAY(ResPosAPI_MenuItem)*ppMenuItems, [in, out] SAFEARRAY(ResPosAPI_ComboMeal)*ppComboMeals, [in, out] ResPosAPI_SvcCharge *pServiceChg, [in, out] ResPosAPI_Discount *pSubTotalDiscount, [in, out] ResPosAPI_TmedDetailItemEx *pTmedDetailEx, [in, out] ResPosAPI_TotalsResponse *pTotalsResponse, [in, out] SAFEARRAY(BSTR)*ppCheckPrintLines, [in, out] SAFEARRAY(BSTR)*ppVoucherOutput)

  Add or change transaction.

- HRESULT AddPaymentBySeat ([in] ResPosAPI_GuestCheck *pGuestCheck, [in] SAFEARRAY(ResPosAPI_SeatPayment)*pCheckSeatPayment, [in, out] SAFEARRAY(BSTR)*ppCheckPrintLines, [in, out] SAFEARRAY(ResPosAPI_SeatSummary)*pSeatResponse, [in, out] SAFEARRAY(ResPosAPI_SeatTmedVoucher)*ppSeatVoucherOutput)

  Add or change transaction.

- HRESULT LockDispatch ([in, out] ResPosAPI_GuestCheck *pGuestCheck, [in] VARIANT_BOOL bLock)

  Lock a check from being dispatched.

- HRESULT AddVouchersOnExistingCheck ([in] ResPosAPI_GuestCheck *pGuestCheck, [in] ResPosAPI_TmedDetailItemEx *pSvtTmedDetail, [in] ResPosAPI_TmedDetailItemEx *pGiftCardTmedDetail, [in, out]SAFEARRAY(ResPosAPI_GiftCardItem)*ppItems)

  Add voucher details on the check.

- HRESULT SetActivityStatus ([in] ResPosAPI_ActivityStatus *pActivity)

  Add activity status detail for the system.

- HRESULT SetActivityStatusList ([in] SAFEARRAY(ResPosAPI_ActivityStatus)*ppActivityList)

  Add a list of activity status details for the system.

- HRESULT ClearAllActivityStatus ()

  Clear All activity status lines in DB.

- HRESULT SetDDCallTimeOnExistingCheck ([in] ResPosAPI_GuestCheck *pGuestCheck, [in] DATE call_begun_tm, [in] DATE call_answer_tm)

  Set call times on delivery dispatch checks.

- HRESULT LockDispatchEx ([in, out] ResPosAPI_GuestCheck *pGuestCheck, [in] VARIANT_BOOL bLock, [in] BSTR appID)

  Lock a check from being dispatched. When using the LockDispatch, the existing Check will be locked and Dispatch operations will be prevented.

Detailed Description

The Main API interface.

This is the interface used for working with POS transactions.

Member Function Documentation

HRESULT CalculateTransactionTotals ([in, out] SAFEARRAY(ResPosAPI_MenuItem)* *ppMenuItems*, [in, out] SAFEARRAY(ResPosAPI_ComboMeal)* *ppComboMeals*, [in, out] ResPosAPI_SvcCharge * *pServiceChg*, [in, out] ResPosAPI_Discount * *pSubTotalDiscount*, [in] LONG *revenueCenterObjectNum*, [in] SHORT *orderType*, [in] LONG *employeeObjectNum*, [in, out] ResPosAPI_TotalsResponse * *pTotalsResponse*)

Calculate Transaction Totals

CalculateTransactionTotals is used to determine the total of the desired transaction, in addition it will validate the basic requirements for posting the desired menu items. Items can be passed together or one by one. No guest check is opened at the target database.

Parameters:

ppMenuItems	**The menu item detail.**
ppComboMeals	The list of combo meals.
pSubTotalDiscount	The subtotal discounts.
revenueCenterObjectNum	Desired revenue Center Object number.
orderType	The order type object number.
employeeObjectNum	The API employee.
pTotalsResponse	Resulting totals response structure.

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

Remarks:
It is highly suggested to use this method to get the total due for the customer. It can also be used to query the API database for menu item prices as the API will return the default price at the desired level in the Menu item definition returned. See the *Menu Item Definition* for more information.

```
ResPosApi::ITransactionServicesPtr pApiTransactionServices = NULL;

// Create an Api transaction services object
hr = CoCreateInstance(ResPosApi::CLSID_TransactionServices,NULL,
   CLSCTX_INPROC_SERVER,ResPosApi::IID_ITransactionServices,
   reinterpret_cast<void**>(&pApiTransactionServices));
```

See Also:

- ResPosApiTypes: ResPosAPI_MenuItem Struct Reference
- ResPosApiTypes: ResPosAPI_ComboMeal Struct Reference
- ResPosApiTypes: ResPosAPI_SvcCharge Struct Reference
- ResPosApiTypes: ResPosAPI_Discount Struct Reference

- ResPosApiTypes: ResPosAPI_TotalsResponse Struct Reference

HRESULT PostTransaction ([in, out] ResPosAPI_GuestCheck * *pGuestCheck*, [in, out] SAFEARRAY(ResPosAPI_MenuItem)* *ppMenuItems*, [in, out] SAFEARRAY(ResPosAPI_ComboMeal)* *ppComboMeals*, [in, out] ResPosAPI_SvcCharge * *pServiceChg*, [in, out] ResPosAPI_Discount * *pSubTotalDiscount*, [in, out] ResPosAPI_TmedDetailItem * *pTmedDetail*, [in, out] ResPosAPI_TotalsResponse * *pTotalsResponse*)

Post transaction to the database

Post Transaction is used to create a new guest check in the RES database.

Parameters:

pGuestCheck	**The check structure to be replaced**
ppMenuItems	Array of menu items to be added to the specified check
ppComboMeals	Array of combo meals to be added to the specified check
pServiceChg	Service Charge to be added to the check.
pSubTotalDiscount	Desired subtotal discount to be added to the specified check
pTmedDetail	The desired tender to be added to the specified check
pTotalsResponse	Resulting totals structure from the new additions

See Also:

- ResPosApiTypes: ResPosAPI_GuestCheck Struct Reference
- ResPosApiTypes: ResPosAPI_MenuItem Struct Reference
- ResPosApiTypes: ResPosAPI_ComboMeal Struct Reference
- ResPosApiTypes: ResPosAPI_SvcCharge Struct Reference
- ResPosApiTypes: ResPosAPI_Discount Struct Reference
- ResPosApiTypes: ResPosAPI_TotalsResponse Struct Reference

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT AddToExistingCheck ([in, out] ResPosAPI_GuestCheck * *pGuestCheck*, [in, out] SAFEARRAY(ResPosAPI_MenuItem)* *ppMenuItems*, [in, out] SAFEARRAY(ResPosAPI_ComboMeal)* *ppComboMeals*, [in, out] ResPosAPI_SvcCharge * *pServiceChg*, [in, out] ResPosAPI_Discount * *pSubTotalDiscount*, [in, out] ResPosAPI_TmedDetailItem * *pTmedDetail*, [in, out] ResPosAPI_TotalsResponse * *pTotalsResponse*)

Add or change transaction

When using the AddToExistingCheck, the Guest Check structure will be interrogated and changed where appropriate. The check sequence number and the check number will not

be changed but the existing Check ID field will always be changed to reflect the new CheckID. Order type will also be changed to reflect the order type passed into the object.

- Fields in the ResPosApi_GuestCheck that are modifiable and will be updated to reflect the new information when using AddToExistingCheck:
    - CheckID
    - CheckTableObjectNum (when supported)
    - CheckOrderType
    - CheckEmployeeObjectNum
    - CheckDateToFire
    - pCheckInfoLines
- Fields in the ResPosApi_GuestCheck that are NOT modifiable when using AddToExistingCheck:
    - CheckRevenueCenterObjectNum
    - CheckSeq
    - CheckNum
- Delayed Order checks:

  The CheckDateToFire element of the guest check structure will allow an order to be delayed on the current business date. When the order is first placed, the order will be delayed based on the CheckDateToFire parameter. The time passed in will be adjusted down by two minutes, so that a POS Operations client will pick up and fire the check on time. This is an internal timing issue and could change in the future. This is just to alert you that the time seen in the database may be slightly different than what was passed to the API.

  Adding to the Delayed order. Delayed order checks can be added to like any other check. If a different fire time is desired, then it should be passed in on the CheckDateToFire element of the check structure. If no change in the fire time is needed, then initialize the CheckDateToFire member to be zero.

- Future Order checks:

  The CheckDateToFire element of the guest check structure along with the CheckStatusBits (Future Order Check [0x10]) will also allow an order to be delayed beyond the current business date. When the order is first placed, the order will be delayed based on the CheckDateToFire parameter. The date/time passed in will be adjusted down by two minutes, so that a POS Operations client will pick up and fire the check on time. This is an internal timing issue and could change in the future. This is just to alert you that the time seen in the database may be slightly different than what was passed to the API.

  Adding to the Future order. Future order checks can be added to like any other check. If a different fire time is desired, then it should be passed in on the CheckDateToFire element of the check structure. If no change in the fire time is needed, then initialize the CheckDateToFire member to be zero.

  The other parameters are considered additions to the existing check and should not include previous items that were added to the check in the PostTransaction call.

Parameters:

pGuestCheck	**The check structure to be replaced**
ppMenuItems-	Array of menu items to be added to the specified check
ppComboMeals	Array of combo meals to be added to the specified check
pServiceChg	Service Charge to be added to the check.
pSubTotalDiscount	Desired subtotal discount to be added to the specified check
pTmedDetail	The desired tender to be added to the specified check
pTotalsResponse	Resulting totals structure from the new additions

See Also:

- ResPosApiTypes: ResPosAPI_GuestCheck Struct Reference
- ResPosApiTypes: ResPosAPI_MenuItem Struct Reference
- ResPosApiTypes: ResPosAPI_ComboMeal Struct Reference
- ResPosApiTypes: ResPosAPI_SvcCharge Struct Reference
- ResPosApiTypes: ResPosAPI_Discount Struct Reference
- ResPosApiTypes: ResPosAPI_TmedDetailItem Struct Reference

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT VoidTransaction ([in, out] ResPosAPI_GuestCheck * *pGuestCheck*)

Void all of the items on the guest check

When using the VoidTransaction, the Guest Check structure will be interrogated and changed where appropriate. The check sequence number and the check number will not be changed but the existing Check ID field will always be changed to reflect the new CheckID. Order type will not be changed when voiding the transaction.

- Fields in the ResPosApi_GuestCheck that are modifiable and will be updated to reflect the new information when using VoidTransaction:
    - CheckID
    - CheckEmployeeObjectNum
    - pCheckInfoLines
- Fields in the ResPosApi_GuestCheck that are NOT modifiable when using AddToExistingCheck:
    - CheckRevenueCenterObjectNum
    - CheckTableObjectNum (when supported)
    - CheckOrderType
    - CheckSeq

- CheckNum
- CheckDateToFire (when supported and only for an existing future order)

Parameters:

pGuestCheck	**The check structure to be Transaction voided**

See Also:

ResPosApiTypes: ResPosAPI_GuestCheck Struct Reference

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT CheckPrintJobStatus ([in, out] SAFEARRAY(LONG)* *ppJobIdList*, [in, out] SAFEARRAY(ResPrintAPI_PrintJobStatus)* *ppJobStatusList*)

Returns the status of a print job

This method allows the user get the status of a single print job base on its status identifier.

Parameters:

ppJobIdList	**Are the IDs of the print jobs**
pStatus	Is a status structure

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT PostTransactionEx ([in, out] ResPosAPI_GuestCheck * *pGuestCheck*, [in, out] SAFEARRAY(ResPosAPI_MenuItem)* *ppMenuItems*, [in, out] SAFEARRAY(ResPosAPI_ComboMeal)* *ppComboMeals*, [in, out] ResPosAPI_SvcCharge * *pServiceChg*, [in, out] ResPosAPI_Discount * *pSubTotalDiscount*, [in, out] ResPosAPI_TmedDetailItemEx * *pTmedDetailEx*, [in, out] ResPosAPI_TotalsResponse * *pTotalsResponse*, [in, out] SAFEARRAY(BSTR)* *ppCheckPrintLines*, [in, out] SAFEARRAY(BSTR)* *ppVoucherOutput*)

Post transaction to the database

Post Transaction is used to create a new guest check in the RES database.

The Printed elements of the check returned have special formatting characters that need to be understood so the elements can be modified to suit the caller. The check is returned in such a way that it can be re-submitted to the RES Print API without modification. See the IGetCheckInfo Interface Reference for more information.

Parameters:

pGuestCheck	**The check structure to be replaced**
ppMenuItems-	Array of menu items to be added to the specified check
ppComboMeals	Array of combo meals to be added to the specified check
pServiceChg	Service Charge to be added to the check.
pSubTotalDiscount	Desired subtotal discount to be added to the specified check
pTmedDetailEx	The desired tender and optionally e-payment information to be added to the specified check
pTotalsResponse	Resulting totals structure from the new additions
ppCheckPrintLines	Raw printed check, can be re-submitted to the Print API
ppVoucherOutput	Raw Credit Voucher (two vouchers may be returned)

See Also:

- ResPosApiTypes: ResPosAPI_GuestCheck Struct Reference
- ResPosApiTypes: ResPosAPI_MenuItem Struct Reference
- ResPosApiTypes: ResPosAPI_ComboMeal Struct Reference
- ResPosApiTypes: ResPosAPI_SvcCharge Struct Reference
- ResPosApiTypes: ResPosAPI_Discount Struct Reference
- ResPosApiTypes: ResPosAPI_TmedDetailItem Struct Reference
- ResPosApiTypes: ResPosAPI_TotalsResponse Struct Reference
- IGetCheckInfo Interface Reference

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT AddToExistingCheckEx ([in, out] ResPosAPI_GuestCheck * *pGuestCheck*, [in, out] SAFEARRAY(ResPosAPI_MenuItem)* *ppMenuItems*, [in, out] SAFEARRAY(ResPosAPI_ComboMeal)* *ppComboMeals*, [in, out] ResPosAPI_SvcCharge * *pServiceChg*, [in, out] ResPosAPI_Discount * *pSubTotalDiscount*, [in, out] ResPosAPI_TmedDetailItemEx * *pTmedDetailEx*, [in, out] ResPosAPI_TotalsResponse * *pTotalsResponse*, [in, out] SAFEARRAY(BSTR)* *ppCheckPrintLines*, [in, out] SAFEARRAY(BSTR)* *ppVoucherOutput*)

Add or change transaction

When using the AddToExistingCheck, the Guest Check structure will be interrogated and changed where appropriate. The check sequence number and the check number will not be changed but the existing Check ID field will always be changed to reflect the new CheckID. Order type will also be changed to reflect the order type passed into the object.

- Fields in the ResPosApi_GuestCheck that are modifiable and will be updated to reflect the new information when using AddToExistingCheck:
    - CheckID
    - CheckTableObjectNum (when supported)
    - CheckOrderType
    - CheckEmployeeObjectNum
    - CheckDateToFire
    - pCheckInfoLines
- Fields in the ResPosApi_GuestCheck that are NOT modifiable when using AddToExistingCheck:
    - CheckRevenueCenterObjectNum
    - CheckSeq
    - CheckNum
- Delayed Order checks:

  The CheckDateToFire element of the guest check structure will allow an order to be delayed on the current business date. When the order is first placed the order will be delayed based on the CheckDateToFire parameter. The time passed in will be adjusted down by two minutes, so that a POS Operations client will pick up and fire the check on time. This is an internal timing issue and could change in the future. This is just to alert you that the time seen in the database may be slightly different than what was passed to the API.

  Adding to the Delayed order. Delayed order checks can be added to like any other check. If a different fire time is desired, then it should be passed in on the CheckDateToFire element of the check structure. If no change in the fire time is needed, then initialize the CheckDateToFire member to be zero.
- Future Order checks:

  The CheckDateToFire element of the guest check structure along with the CheckStatusBits (Future Order Check [0x10]) will also allow an order to be delayed beyond the current business date. When the order is first placed, the order will be delayed based on the CheckDateToFire parameter. The date/time passed in will be adjusted down by two minutes, so that a POS Operations client will pick up and fire the check on time. This is an internal timing issue and could change in the future. This is just to alert you that the time seen in the database may be slightly different than what was passed to the API.

  Adding to the Future order. Future order checks can be added to like any other check. If a different fire time is desired, then it should be passed in on the CheckDateToFire element of the check structure. If no change in the fire time is needed, then initialize the CheckDateToFire member to be zero.

  The other parameters are considered additions to the existing check and should not include previous items that were added to the check in the PostTransaction call.
- The Printed elements of the check returned have special formatting characters that need to be understood so the elements can be modified to suit the caller. The check is returned in such a way that it can be re-submitted to the RES Print API without modification. See the IGetCheckInfo Interface Reference for more information.

Parameters:

pGuestCheck	**The check structure to be replaced**
ppMenuItems-	Array of menu items to be added to the specified check
ppComboMeals	Array of combo meals to be added to the specified check
pServiceChg	Service Charge to be added to the check.
pSubTotalDiscount	Desired subtotal discount to be added to the specified check
pTmedDetailEx	The desired tender and optionally e-payment information to be added to the specified check
pTotalsResponse	Resulting totals structure from the new additions
ppVoucherOutput	Array of strings
ppCheckPrintLines	Raw printed check, can be re-submitted to the Print API
ppVoucherOutput	Raw Credit Voucher (two vouchers may be returned)

See Also:

- ResPosApiTypes: ResPosAPI_GuestCheck Struct Reference
- ResPosApiTypes: ResPosAPI_MenuItem Struct Reference
- ResPosApiTypes: ResPosAPI_ComboMeal Struct Reference
- ResPosApiTypes: ResPosAPI_SvcCharge Struct Reference
- ResPosApiTypes: ResPosAPI_Discount Struct Reference
- ResPosApiTypes: ResPosAPI_TmedDetailItem Struct Reference
- ResPosApiTypes: ResPosAPI_TotalsResponse Struct Reference
- IGetCheckInfo Interface Reference

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT AddPaymentBySeat ([in] ResPosAPI_GuestCheck * *pGuestCheck*, [in] SAFEARRAY(ResPosAPI_SeatPayment)* *pCheckSeatPayment*, [in, out] SAFEARRAY(BSTR)* *ppCheckPrintLines*, [in, out] SAFEARRAY(ResPosAPI_SeatSummary)* *pSeatResponse*, [in, out] SAFEARRAY(ResPosAPI_SeatTmedVoucher)* *ppSeatVoucherOutput*)

Add or change transaction.

The ResPosAPI_SeatSummary structure is passed as out parameter to enable access to the check seat total information.

The ResPosAPI_SeatSummary structure is passed as out parameter to enable access to the check seat total information.

This function will not generate a print job to ROD.

Parameters:

pGuestCheck	**The check structure to be replaced**
pCheckSeatPayment-	Array of seat tender media items
ppCheckPrintLines	Array of Printed Guest Check Lines. This may be in 32 or 40 column format.
pSeatResponse	Array of seat response
ppSeatVoucherOutput	Array of voucher lines per payment per seat. If a value is passed in this parameter it will be added as voucher addendum.

See Also:

- ResPosApiTypes: ResPosAPI_GuestCheck Struct Reference
- ResPosApiTypes: ResPosAPI_SeatPayment Struct Reference
- ResPosApiTypes: ResPosAPI_SeatSummary Struct Reference

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT LockDispatch ([in, out] ResPosAPI_GuestCheck * *pGuestCheck*, [in] VARIANT_BOOL *bLock*)

Lock a check from being dispatched.

When using the LockDispatch, the existing Check will be locked and Dispatch operations will be prevented.

Parameters:

pGuestCheck	**The check structure to be locked from being dispatched**

See Also:

ResPosApiTypes: ResPosAPI_GuestCheck Struct Reference

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT AddVouchersOnExistingCheck ([in] ResPosAPI_GuestCheck * *pGuestCheck*, [in] ResPosAPI_TmedDetailItemEx * *pSvtTmedDetail*, [in] ResPosAPI_TmedDetailItemEx * *pGiftCardTmedDetail*, [in, out] SAFEARRAY(ResPosAPI_GiftCardItem)* *ppItems*)

Add voucher details on the check

When using the AddVouchersOnExistingCheck, the existing Check will be updated with new voucher details.

Parameters:

pSvtTmedDetailEx	**Used to post changes to DB (service total tmed)**
pGiftCardTmedDetail	Used to get tender options like should the account number be masked
ppItems	A list of gift card items with vouchers to add to the check.
pSvtTmedDetailEx	Used to post changes to DB (service total tmed)

See Also:

ResPosApiTypes: ResPosAPI_GiftCardItem Struct Reference

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT SetActivityStatus ([in] ResPosAPI_ActivityStatus * *pActivity*)

Add activity status detail for the system

Parameters:

pActivity	**Used to post changes to DB (activity status detail)**

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT SetActivityStatusList ([in] SAFEARRAY(ResPosAPI_ActivityStatus)* *ppActivityList*)

Add a list of activity status details for the system

Parameters:

ppActivityList	**Used to post changes to DB (a list of activity status details)**

NOTE:

Before posting to the DB, every activity_type for this origin is reset. The API still makes separate calls to the DB to clear and set the new status for the given origin. Make sure when using this function that the origin of the activity lines are not the same, because it will be overridden by the last one.

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT ClearAllActivityStatus ()

Clear All activity status lines in DB

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT SetDDCallTimeOnExistingCheck ([in] ResPosAPI_GuestCheck * *pGuestCheck*, [in] DATE *call_begun_tm*, [in] DATE *call_answer_tm*)

Set call times on delivery dispatch checks

Parameters:

pGuestCheck	**The check structure to have the call times updated**
call_begun_tm	The time indicating when a patron wanted to place an order
call_answer_tm	The time indicating when a restaurant employee responded to a call
pActivity	Used to post changes to DB (activity status detail)

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

HRESULT LockDispatchEx ([in, out] ResPosAPI_GuestCheck * *pGuestCheck*, [in] VARIANT_BOOL *bLock*, [in] BSTR *appID*)

Lock a check from being dispatched. When using the LockDispatch, the existing Check will be locked and Dispatch operations will be prevented

Parameters:

pGuestCheck	**the check structure to be locked from being dispatched**
bLock	flag to indicate if the check is to be locked or unlocked
appID	string to indicate the application which locked the check. Limited to 16 characters.

See Also:

ResPosApiTypes: ResPosAPI_GuestCheck Struct Reference

Returns:

S_OK for success
Error HRESULT for error (see API Return Codes)

> **NOTE:**
>
> An error will be returned if the check is already locked and another lock is attempted from different appID.

ResPosAPI_ActivityStatus Struct Reference

Structure that defines an activity status

Public Attributes

- EActivityStatusType activity_type
  Activity status type (see enum EActivityStatusType)
- BSTR origin
  Text field with a limit of 16 chars defining the origin of the activity
- BSTR status
  Text field with a limit of 32 chars defining the status text
- DATE status_change_tm
  When the status change occurred?
- SHORT expire_seconds
  Set this to number of seconds when the message will be cleared automatically by the system.

Detailed Description

Structure that defines an activity status

ResPosAPI_CheckSummary Struct Reference

Structure that defines a current open guest check in the RES 3700 system.

Public Attributes

- long CheckSeq
  [out] CheckSeq is an out parameter telling the user the check sequence number of the check.
- long CheckNum
  [out] CheckNum is an out parameter, the check number range can only be specified by User Workstation or Revenue Center in the RES database.
- long CheckEmployeeObjectNum
  [out] CheckEmployeeObjectNum refers to the API check and transaction employee, this will be a configured employee in the RES database
- long CheckRevenueCenterObjectNum
  [out] CheckRevenueCenterObjectNum is the revenue center this check is currently active.
- long CheckLastWorkstationOwner
  [out] CheckLastWorkstationOwner is the last workstation object that owned this check.
- long CheckCurrentlyOpenOnWorkstation
  [out] CheckCurrentlyOpenOnWorkstation is the current workstation object that owns this check.
- long CheckTableObjectNum
  [out] CheckTableObjectNum the check on the table specified
- SHORT CheckTableGroup
  [out] CheckTableGroup, describes the table group
- SHORT CheckOrderType
  [out] CheckOrderType, describes "Eat In", "Take Out", this will be a configured order type in the RES database
- BSTR CheckID
  [out] CheckId specifies the Check name. Duplicate Check names on open checks are not allowed.
- BSTR CheckTotalDue
  [out] CheckTotalDue specifies the amount due on the open check.
- DATE CheckLastServiceTime
  [out] Last Service Time describes the last time this check was submitted to the transaction system
- DATE CheckOpenTime
  [out] CheckOpenTime describes when the Check was opened on the transaction system.
- DATE CheckAutoFireTime
  [out] CheckAutoFireTime describes when this Check will be fired.
- VARIANT_BOOL CheckInTraining
  [out] CheckTraining describes the training state of this check.
- VARIANT_BOOL CheckInsufficientBeverage
  [out] CheckInsufficientBeverage describes the status of this check as having insufficient beverage.

- VARIANT_BOOL CheckTransferedToDriver
  [out] CheckTransferedToDriver describes the status of this check as having been assigned to a driver.
- VARIANT_BOOL CheckIsDelayedOrder
  [out] CheckIsDelayedOrder describes the status of this check as being a Delayed Order
- VARIANT_BOOL CheckIsFutureOrder
  [out] CheckTransferedToDriver describes the status of this check as having been assigned to a driver.

Detailed Description

Structure that defines a current open guest check in the RES 3700 system. This structure always refers to a currently Open Check on the point of sale system.

ResPosAPI_CheckSummaryWithSeats Struct Reference

Structure that defines a guest check in the RES 3700 system (open or close). Includes summary information by seat (totals and payments).

Public Member Functions

SAFEARRAY

- (ResPosApiTypes: ResPosAPI_SeatPayment Struct Reference) Seats
  [out] Seats describes check's seats total and payment

Public Attributes

- long CheckSeq
  [out] CheckSeq is an out parameter telling the user the check sequence number of the check.
- long CheckNum
  [out] CheckNum is an out parameter, the check number range can only be specified by User Workstation or Revenue Center in the RES database.
- long CheckEmployeeObjectNum
  [out] CheckEmployeeObjectNum refers to the API check and transaction employee, this will be a configured employee in the RES database
- long CheckRevenueCenterObjectNum
  [out] CheckRevenueCenterObjectNum is the revenue center this check is currently active.
- long CheckLastWorkstationOwner
  [out] CheckLastWorkstationOwner is the last workstation object that owned this check.
- long CheckCurrentlyOpenOnWorkstation
  [out] CheckCurrentlyOpenOnWorkstation is the current workstation object that owns this check.

- long CheckTableObjectNum
  [out] CheckTableObjectNum the check on the table specified
- SHORT CheckTableGroup
  [out] CheckTableGroup, describes the table group
- SHORT CheckOrderType
  [out] CheckOrderType, describes "Eat In", "Take Out", this will be a configured order type in the RES database
- BSTR CheckID
  [out] CheckId specifies the Check name. Duplicate Check names on open checks are not allowed.
- BSTR CheckTotalDue
  [out] CheckTotalDue specifies the amount due on the open check.
- BSTR CheckTotalPayment
  [out] CheckTotalPayment specifies the amount paid on the check.
- DATE CheckLastServiceTime
  [out] Last Service Time describes the last time this check was submitted to the transaction system
- DATE CheckOpenTime
  [out] CheckOpenTime describes when the Check was opened on the transaction system.
- DATE CheckAutoFireTime
  [out] CheckAutoFireTime describes when this Check will be fired.
- VARIANT_BOOL CheckInTraining
  [out] CheckTraining describes the training state of this check.
- VARIANT_BOOL CheckInsufficientBeverage
  [out] CheckInsufficientBeverage describes the status of this check as having insufficient beverage.
- VARIANT_BOOL CheckTransferedToDriver
  [out] CheckTransferedToDriver describes the status of this check as having been assigned to a driver.
- VARIANT_BOOL CheckIsDelayedOrder
  [out] CheckIsDelayedOrder describes the status of this check as being a Delayed Order
- VARIANT_BOOL CheckIsFutureOrder
  [out] CheckTransferedToDriver describes the status of this check as having been assigned to a driver.
- VARIANT_BOOL CheckIsClosed
  [out] CheckIsClosed describes the status of this check as having been closed.

Detailed Description

Structure that defines a guest check in the RES 3700 system (open or close). Includes summary information by seat also (totals and payments).

ResPosAPI_ComboMeal Struct Reference

Base Combo Meal Definition.

Public Member Functions

SAFEARRAY

(ResPosAPI_MenuItem Struct Reference) SideItems

Combo Meal Side Items (ex FF or Coke). ***The array is passed as a SAFEARRAY.***

Public Attributes

- ResPosAPI_MenuItem Struct Reference ComboMealMenuItem

  Combo Meal Menu Item (ex Burger Combo)
- long ComboMealObjectNum
  Describes this item as the Combo Group Number.
- ResPosAPI_MenuItem Struct Reference ComboMealMainItem

  Combo Meal Main Item (ex Hamburger)

Detailed Description

Base Combo Meal Definition

When ordering Combo meals, we are very strict in checking all the combo meal linkage. The combo meal menu item passed must be linked to the combo meal object number, additionally the side items passed must be linked correctly to the combo meal as defined in the target database. This means sides must also be passed in order. The following select statement may be used as a guide: select * from micros.combo_side_def left outer join micros.combo_def order by obj_num

ResPosAPI_CreditCard Struct Reference

Credit Card Information

Public Attributes

- BSTR CCAcctNumber
  Account Number [in, out].
- DATE CCExpirationDate
  Expiration date [in, out].
- BSTR CCAuthorizationCode
  Authorization Code [in, out].
- DATE CCStartDate
- SHORT CCIssueNumber
  CCIssueNumber [in] not modified.
- BSTR CCTrack1Data
  Track 1 Data [in, out].

- BSTR CCTrack2Data
  Track 2 Data [in, out].
- BSTR CCTrack3Data
  Track 3 Data [in, out].
- BSTR CCBaseAmount
  Base Amount [in] not modified.
- BSTR CCTipAmount
  Tip Amount [in] not modified.
- BSTR CCCVVNumber
  CVV Number [in, out].
- BSTR CCAddressVerification
  Address Verification [in, out].

Detailed Description

Credit Card Information
For future use.

Member Data Documentation

BSTR CCAcctNumber

Account Number [in, out].

The Account number is mainly used for manual entry accounts. ***This data will be destroyed by the API and not returned to the caller!***

DATE CCExpirationDate

Expiration date [in, out].

The expiration date is mainly used for manual entry accounts. ***This data will be destroyed by the API and not returned to the caller!***

BSTR CCAuthorizationCode

Authorization Code [in, out].

The authorization code is normally only used as an out parameter, however if the client populates this field as an [in] parameter, then the API will treat the credit payment as pre-authorized, and place the authorization code on the check.

DATE CCStartDate

CCStartDate [in] not modified field required by some electronic payment processors.

SHORT CCIssueNumber

CCIssueNumber [in] not modified field required by some electronic payment processors.

BSTR CCTrack1Data

Track 1 Data [in, out].

Magnetic Card Track 1. ***This data will be destroyed by the API and not returned to the caller!***

BSTR CCTrack2Data

Track 2 Data [in, out].

Magnetic Card Track 2. ***This data will be destroyed by the API and not returned to the caller!***

BSTR CCTrack3Data

Track 3 Data [in, out].

Magnetic Card Track 3. ***This data will be destroyed by the API and not returned to the caller!***

BSTR CCBaseAmount

Base Amount [in] not modified.

The base amount is the amount of payment desired without tip or cash back. Base Amount = Total Amount - Tip Amount

BSTR CCTipAmount

Tip Amount [in] not modified.

The desired Tip to be applied to the payment.

BSTR CCCVVNumber

CVV Number [in, out].

Card Verification Value. ***This data will be destroyed by the API and not returned to the caller!***

BSTR CCAddressVerification

Address Verification [in, out].

Address Verification String. ***This data will be destroyed by the API and not returned to the caller!***

ResPosAPI_Discount Struct Reference

Structure used to represent a discount in Transaction Services.

Public Attributes

- long DiscObjectNum
  Discount Object Number
- BSTR DiscAmountOrPercent
  Discount Amount or Percentage
- BSTR DiscReference
  Optional Reference entry

Detailed Description

**Structure used to represent a discount in Transaction Services.
These fields are all mapped to the micros database query SELECT * FROM "micros"."dsvc_def" where type = 'D'**

ResPosAPI_EPayment Struct Reference

Advanced Electronic Payment Structure

Public Attributes

- EPaymentDirective AccountDataSource
  EPaymentDirective [in] not modified.
- EAccountDataSource AccountDataSource
  EAccountDataSource [in] not modified.
- EAccountType AccountType
  EAccountDataSource [in] not modified.
- BSTR AcctNumber
  Account Number [in, out].
- DATE ExpirationDate
  Expiration date [in, out].
- BSTR AuthorizationCode
  Authorization Code [in, out].
- DATE StartDate
  Start date [in] not modified.
- SHORT IssueNumber
  Issue Number [in] not modified.
- BSTR Track1Data
  Track 1 Data [in, out].
- BSTR Track2Data
  Track 2 Data [in, out].
- BSTR Track3Data
  Track 3 Data [in, out].
- BSTR BaseAmount
  Base Amount [in] not modified.
- BSTR TipAmount
  Tip Amount [in] not modified.
- BSTR CashBackAmount
  Cash Back Amount [in] not modified.
- BSTR KeySerialNum
  Key Serial Number [in] not modified.
- BSTR DeviceId
  Device Identifier [in] not modified.

- BSTR PinBlock
  Pin Number Encrypted by caller [in] not modified.
- BSTR CVVNumber
  CVV Number [in, out].
- BSTR AddressVerificationBSTR AddressVerificationBSTR AddressVerification
  Address Verification [in, out].
- BSTR InterfaceName
  Interface name [in].
- BSTR SvcResponse
  Stored Value Card Message Response.
- BSTR SvcAccountType
  Stored Value Account [in].

Detailed Description

Advanced Electronic Payment Structure

The advanced electronic payment information is available for use when advanced electronic payment is required.
There are special fields used for payment amount information when using the electronic payment structure.

- **BaseAmount**
- **TipAmount**
- **CashBackAmount**

When the BaseAmount or the TipAmount or the CashBackAmount field(s) are used it will override the ResPosAPI_TmedDetailItemEx.TmedPartialPayment amount, meaning the TmedPartialPayment will be ignored.
The BaseAmount is used for defining how much of the existing check total you want to pay. The TipAmount and the CashBackAmount fields will be added and authorized (if configured) and paid automatically by the API.

Examples

Assume a check total of 50.00.
Example 1

- **BaseAmount = "25.00"**
- **TipAmount = "5.00"**
- **CashBackAmount = "0.00"**

The API interprets this request as pay 25.00 and add a 5.00 charge tip. The total payment amount charged to the credit card will be 30.00 (25.00 payment plus 5.00 charge tip).
Example 2

- **BaseAmount = "0.00"**
- **TipAmount = "5.00"**
- **CashBackAmount = "0.00"**

The API interprets this request as pay 0.00 and add a 5.00 charge tip. The total payment amount charged to the credit card will be 5.00 (0.00 payment plus 5.00 charge tip).
Example 3

- **BaseAmount = "55.00"**
- **TipAmount = "10.00"**
- **CashBackAmount = "0.00"**

The API interprets this request as pay 50.00 and add a 10.00 charge tip. In this case the developer has tried to apply a 55.00 payment for a 50.00 check. The 55.00 payment is ignored and internally modified by the API to correct the overpayment and set the BaseAmount to 50.00. The total payment amount charged to the credit card will be 60.00 (50.00 payment plus 10.00 charge tip).
Example 4

- **BaseAmount = "55.00"**
- **TipAmount = "5.00"**
- **CashBackAmount = "7.00"**

The API interprets this request as pay 50.00 and add a 5.00 charge tip and 7.00 cash back. In this case the developer has tried to apply a 55.00 payment for a 50.00 check. This is modified by the API to correct the overpayment and push the amount back to 50.00. The total payment amount charged to the credit card will be 62.00 (50.00 payment plus 5.00 charge tip plus 7.00 cash back).

Member Data Documentation

EPaymentDirective PaymentCommand

EPaymentDirective [in] not modified.

Defines what action is desired. Valid values include:

- NO_E_PAYMENT
- AUTHORIZE_ONLY
- AUTHORIZE_AND_PAY
- DEBIT_AUTHORIZE_ONLY
- DEBIT_AUTHORIZE_AND_PAY
- STORED_VALUE_CARD_AUTHORIZE
- STORED_VALUE_CARD_REDEEM
- GET_DEBIT_CANCEL_VOUCHER

EAccountDataSource AccountDataSource

EAccountDataSource [in] not modified.

Defines what account data source should be used for the transaction:

- SOURCE_UNDEFINED
- RFID_TRACK_DATA_RULES
- RFID_M_CHIP_RULES
- MANUALLY_KEYED_TRACK_1_CAPABLE
- MANUALLY_KEYED_TRACK_2_CAPABLE
- MANUALLY_KEYED_NO_CARD_READER

EAccountType AccountType

EAccountDataSource [in] not modified.

Defines what account type should be used for the transaction:

- ACCOUNT_TYPE_UNDEFINED,
- CHECKING,
- SAVINGS,

BSTR AcctNumber

Account Number [in, out].

The Account number is mainly used for manual entry accounts. ***This data will be destroyed by the API and not returned to the caller!***

DATE ExpirationDate

Expiration date [in, out].

The expiration date is mainly used for manual entry accounts. ***This data will be destroyed by the API and not returned to the caller!***

BSTR AuthorizationCode

Authorization Code [in, out].

The authorization code is normally only used as an out parameter, however if the client populates this field as an [in] parameter, then the API will treat the credit payment as pre-authorized, and place the authorization code on the check.

DATE StartDate

Start date [in] not modified.

Field required by some processors

SHORT IssueNumber

Issue Number [in] not modified.

Field required by some electronic payment processors

BSTR Track1Data

Track 1 Data [in, out].

Magnetic Card Track 1. ***This data will be destroyed by the API and not returned to the caller!***

BSTR Track2Data

Track 2 Data [in, out].

Magnetic Card Track 2. ***This data will be destroyed by the API and not returned to the caller!***

BSTR Track3Data

Track 3 Data [in, out].

Magnetic Card Track 3. ***This data will be destroyed by the API and not returned to the caller!***

BSTR BaseAmount

Base Amount [in] not modified.

The base amount is the amount of payment desired without tip or cash back Base Amount = Total Amount - Tip Amount - Cash Back Amount. This could be changed internally by the API, but this field will not be modified.

BSTR TipAmount

Tip Amount [in] not modified.

The desired Tip to be applied to the payment

BSTR CashBackAmount

Cash Back Amount [in] not modified.

The desired cash back to be applied to the payment

BSTR KeySerialNum

Key Serial Number [in] not modified.

Debit Key Serial Number (max 20)

BSTR DeviceId

Device Identifier [in] not modified.

Device Identifier (max 9 characters)

BSTR PinBlock

Pin Number Encrypted by caller [in] not modified.

Pin Number Encrypted (never plain text). Used only with Debit (max 16)

BSTR CVVNumber

CVV Number [in, out].

This data will be destroyed by the API and not returned to the caller!

The Card Verification Value (CVV) code, is the number used to verify the card is in possession of the payer. It may be required for some credit transactions that include keyed account information. A blank CVV value (one or more spaces) may be used to send a CVV status of 'CVV intentionally not provided' in the authorization request.

BSTR AddressVerification

Address Verification [in, out].

This data will be destroyed by the API and not returned to the caller!

Address Verification, another security measure (usually blank for most processors), may be required for some credit transactions that include keyed account information. Both the postal code and street address may be included separated by a '|' character.

BSTR InterfaceName

Interface name [in].

The interface name is a unique identifier for interfaces stored in the RES 3700 database. Many gift card processors will report this name in their transaction manifest, so that the database will not need to be contacted.

SQL call "`SELECT msg_interface_name FROM micros.interface_def`".

BSTR SvcResponse

Stored Value Card Message Response.

If a stored value card operation is rejected, a descriptive message may be returned in this field.

BSTR SvcAccountType

Stored Value Account [in].

Max 32 characters

ResPosAPI_GiftCardItem Struct Reference

Structure that defines a gift card item on a check.

Public Attributes

- ResPosApi_GiftCardItemDef Struct Reference
  [out] Gift card item definition
- long dtlId
  [out] Check detail id representing a gift card operation (activation, reload, etc.)
- VARIANT_BOOL isVoid
  [out] Gift card Item void flag
- BSTR amount
  [out] Gift card Item amount
- BSTR account
  Account buffer
- BSTR voucher
  Voucher buffer.
- int action
  Result action (0 - add voucher detail, 1 - void detail.

Detailed Description

Structure that defines a gift card item on a check.

ResPosApi_GiftCardItemDef Struct Reference

Structure that defines a gift card item.

Public Attributes

- long num
  [out] Check detail object number (if type is menu item) or sequence (if type is not menu item) representing a gift card operation (activation, reload, etc.)
- ResPosAPI_EDtlType dtlType
  [out] Check detail type (MI, SVC) representing a gift card operation (activation, reload, etc.)

Detailed Description

Structure that defines a gift card item.

ResPosAPI_GuestCheck Struct Reference

Guest Check structure

Public Member Functions

- SAFEARRAY (BSTR) pCheckInfoLines
  The array is passed as a SAFEARRAY.
- SAFEARRAY (long) pPrintJobIds

Public Attributes

- BSTR CheckID
  [in,out] CheckId is in, out parameter specifying the Check name. Duplicate Check names on open checks are not allowed.
- long CheckTableObjectNum
  [in] CheckTableObjectNum is not yet supported, this will eventually open a check on the table specified
- long CheckRevenueCenterObjectNum
  [in] CheckRevenueCenterObjectNum is the desired revenue center this check should be created in.
- SHORT CheckOrderType
  [in] CheckOrderType is the desired order type, like "Eat In", "Take Out", this will be a configured order type in the RES database
- long CheckEmployeeObjectNum
  [in] CheckEmployeeSeq refers to the API check and transaction employee, this will be a configured employee in the RES database
- long CheckSeq
  [out] CheckSeq is on in, out parameter telling the user the check sequence number of the newly created check. This is used as an in parameter when adding items to an existing check.
- long CheckNum
  [out] CheckNum is an out parameter, the check number range can only be specified by User Workstation or Revenue Center in the RES database.
- DATE CheckDateToFire
  [in] CheckDateToFire will permit an order to be delayed or fired on a future date.
- SHORT CheckGuestCount
  [in] CheckGuestCount is a user supplied guest count, similar to prompt in POS Operations
- long CheckStatusBits
  [in] 32 Bits of Check status identifiers, think Rush Order or VIP

Detailed Description

Guest Check structure

The guest check structure is a collection of elements that are passed as an in, out parameter. This shared structure is used to communicate key elements of the transaction to the API and for the API to return key elements to the API user.

Member Function Documentation

SAFEARRAY (long)

[out] Array of print job ID that resulted from the transaction. **The array is passed as a SAFEARRAY.**

Member Data Documentation

DATE CheckDateToFire

[in] CheckDateToFire will permit an order to be delayed or fired on a future date.

> **NOTE:**
>
> The CheckStatusBits (0x10) controls the check being fired on the current business date or a future date.
> When cleared, the order will be fired on the current business date.
> When set, the order will be fired on a future date.
> Up to 2038-01-19 03:14:08 (UTC)

long CheckStatusBits

[in] 32 Bits of Check status identifiers, think Rush Order or VIP

```
enum GuestCheckStatusBits
{
   RUSH_ORDER = 0,
   HIGH_PRIORITY_CHECK,
   EMPLOYEE_MUST_OWN_CHECK,
   ALLOW_PARTIAL_AUTH,
   FUTURE_CHECK,
   DONOT_CLOSE_CHECK,
   KEEP_CHK_CLOSED_DATE_TIME,
   SUPPRESS_CHK_PRINT,
   SUPPRESS_VOUCHER_PRINT,
   REMOVE_AUTO_DISCOUNTS,
}GuestCheckStatusBits;
```

The definition of each bit position is as follows:

1. Rush Order
2. VIP Transaction
3. Employee must own the check that is being picked up
4. Allow Partial Auth
5. Future Order Check

6. Do not close the check even if paid in full
7. Keep check closed date time if reopen closed check
8. Do not print a check receipt
9. Do not print a voucher
10. Remove automatic discounts
11. Thru 32 are unused

ResPosAPI_MenuItem Struct Reference

Menu Item and Modifiers (Condiments)

Public Member Functions

SAFEARRAY (ResPosAPI_MenuItemDefinition Struct Reference) Condiments

Condiment array. ***The array is passed as a SAFEARRAY***

Public Attributes

ResPosAPI_MenuItemDefinition Struct Reference, ResPosAPI_MenuItem Struct Reference

Describes this item as a generic Main Item.

Detailed Description

Menu Item and Modifiers (Condiments)

The menu item is comprised of the main item and an array of condiments. An example may be Cheeseburger (main item), well, extra, pickles (condiment array).

Menu Item Reference Entry:

The Menu Item Reference entry field may be used to convey additional information about a menu item for posting into mi_dtl records.

Allowed Condiment**:**

The API code will add all condiments as if they were 'Required'. Because the system ignores the Print Group of 'Required' condiments when sorting, you need to be able to add an 'Allowed' condiment when you want a condiment to be sorted by print group. In order to add an 'Allowed' condiment, you need to provide a reference entry that contains the following information:
<ATTRIB>ALLOWED_CONDIMENT</ATTRIB>
A reference for a 'Required' condiment entry looks like:
"Hello World"
A reference for an 'Allowed' condiment entry looks like:
"Hello World<ATTRIB>ALLOWED_CONDIMENT</ATTRIB>"
A reference entry for an 'Allowed' condiment that is not intended to add a reference line looks like:
"<ATTRIB>ALLOWED_CONDIMENT</ATTRIB>"
Ordering Module Fields:
The mi_dtl database contains 8 columns that are populated by the RES Ordering Module (OM), that are not processed by the API code. A custom ordering application may wish to populate these columns with data, for use in reporting or compatibility when picking up a check to modify with the RES Ordering Module. All of these fields are optional and may be entered in any order on the reference line, but the tags ARE case-sensitive. Any invalid tag entries will be ignored and any "non-tagged" data in the Menu Item Reference field will be considered as the menu item reference data to display/print:

<omType>n</omType>

The Ordering Module Type (numeric, 1-5). The OM Type of each detail will determine the fields required to maintain compatibility with the RES Ordering Module. The OM Type values are presented with their respective required data fields below:
OM_BASE_RECORD_TYPE (1): The OM Base Record Type MUST be the first record in a group of details to be compatible with the RES Ordering Module. The Base Record type represents the Size/Base of the item (e.g., Large Thin Crust). ***Required fields : <omNumber>,<omBaseSeq>***
OM_SECTION_RECORD_TYPE (2): An OM Section Record type indicates to what portion of the item the next detail(s) will apply (e.g., Half, Whole, etc.). ***Required fields : <omNumber>,<omBaseSeq>,<omSectionInfoSeq>***
OM_SPECIALTY_RECORD_TYPE (3): The OM Specialty Record is used to order a known set of toppings on the base item to prepare. ***Required fields : <omNumber>,<omBaseSeq>,<omSectionInfoSeq>,<omSpecialtySeq>***

OM_TOPPING_RECORD_TYPE (4): An OM Topping Record is a record containing a topping menu item and modifier, for customize the item to prepare. (e.g., Double Mushroom, No Anchovies, etc.). Required fields ***: <omNumber>,<omBaseSeq>,<omSectionInfoSeq>, <omSpecialtySeq>,<omToppingSeq>,<omMiToppingSeq>,<omModifierSeq>***

OM_CONTAINER_RECORD_TYPE (5): A container record <optional, for inventory tracking>, must be the LAST record in an OM Item group. It provides a means to

count the container used to package the item (e.g., Large Pizza Box). *Required fields : <omNumber>,<omBaseSeq>*
<omNumber>n</omNumber>
This number is used to identify each item within an order processed by the OM. For example, if an order contained 1 Large Pan Pizza w/Pepperoni and Green Peppers and 1 Medium Thin Pizza w/Ham and Pineapple, all detail entries associated with the Large Pan Pizza would have <omNumber>1</omNumber> and all detail entries of the Medium Thin Pizza would have <omNumber>2</omNumber>.
<omBaseSeq>n</omBaseSeq>
The sequence number of the om_base_def record. The OM Base definition represents the basic starting ingredient of the OM item (e.g., Large Deep Dish, 6" Whole Wheat, etc.). Each menu item can link to one and only one omBase record, and the omBaseSeq may be determined using the SQL Query:
SELECT base_seq FROM "micros"."om_base_def" WHERE mi_seq = [mi sequence].
NOTE: The [mi sequence] value should be the "mi_def.mi_seq" field matching the parent menu item detail record for this item group.
<omSectionInfoSeq>n</omSectionInfoSeq>

The sequence number of the om_section_info_def record. The OM Section Info definition identifies "section" of the item is being modified (e.g., Left half of pizza, Right half of sub, etc.). The omSectionInfoSeq may be derived using the SQL Query:
SELECT section_info_seq FROM "micros"."om_section_info_def" WHERE mi_seq = [mi sequence] AND section_seq = [section sequence] AND obj_num = [section number]
NOTE: The user-configurable fields:

1. **[mi sequence] value should be the "mi_def.mi_seq" field matching the section detail of this modifier group,**
2. **[section sequence] identifies the section type from "om_sec_def.section_seq" (e.g., Round Half, Square Half, etc.) and**
3. **[section number] refers to the specific section (e.g., Left half, Right half, etc.) must be provided by the application, based on ordering details.**

<omSpecialtySeq>n</omSpecialtySeq>
The sequence number of the om_specialty_def record. The OM Specialty definition represents the primary topping(s)/ingredient(s) of the item to prepare (e.g., Meat Lover's, Cheese, etc.). The omSpecialtySeq may be obtained using the following SQL Query:
SELECT specialty_seq FROM "micros"."om_specialty_def" WHERE

NOTE: The user-configurable fields:

1. **[base sequence] refers to the <omBase> value from the Parent item in this item group,**
2. **[section sequence] identifies the section type from "om_sec_def.section_seq" (e.g., Round Half, Square Half, etc.) and**
3. **[section number] refers to the specific section (e.g., Left half, Right half, etc.) must be provided by the application, based on ordering details.**

<omToppingSeq>n</omToppingSeq>
The sequence number of the om_topping_type_def record.
<omToppingMiSeq>n</omToppingMiSeq>
The sequence number of the om_topping_mi_def record.

<omModifierSeq>n</omModifierSeq>
The sequence number of the om_modifier_def record.

ResPosAPI_MenuItemDefinition Struct Reference

The Base Menu item Definition

Public Attributes

- long MiObjectNum
  Menu Item Object number in the Micros Database.
- long MiMenuLevel
  Menu Level. Must be a Value between 1-10. The level can be used for default pricing by the API.
- BSTR MiOverridePrice
  (Optional parameter) String representing Override Price. This field can be left empty if the menu level default price is desired. If left NULL the price will be returned to the caller here
- BSTR MiWeight
  (Optional parameter) Describes the weight of an item
- BSTR MiReference
  (Optional parameter) This will add a reference entry to the menu item, but only uses the first 20 characters
- ResPosAPI_Discount Struct Reference ItemDiscount
  (Optional parameter) This is for application of an item discount

Detailed Description

The Base Menu item Definition

Before using the ResPosAPI_MenuItemDefinition, it is highly suggested that each field is initialized properly. Leaving this object in in an uninitialized state can lead to unexpected results.
The MiOverridePrice can be used to override the price of the menu item, if the default price is desired, leave this NULL and the POS API will return the default price used for the totals calculation in the MiOverridePrice BSTR.
Weighed items will not remove the tare weight, it is assumed the weight passed is the actual weight.

ResPosAPI_SeatPayment Struct Reference

Structure that defines a check seat payment information

Public Member Functions

SAFEARRAY

(ResPosAPI_TmedDetailItemEx Struct Reference) pSeatPayments
Seat's payment detail

Detailed Description

Structure that defines a check seat payment information.

ResPosAPI_SeatSummary Struct Reference

Structure that defines a check seat total information.

Public Attributes

- long SeatNum
  Check's seat number.
- ResPosAPI_TotalsResponse Struct Reference SeatTotals
  Seat's total
- BSTR SeatTotalPayment
  Seat's Total Payment.

Detailed Description

Structure that defines a check's seat total information.

ResPosAPI_SeatTmedVoucher Struct Reference

Structure that defines a voucher information by seat.

Public Member Functions

SAFEARRAY

(BSTR) ppVoucherOutput
Seat's tender media voucher

Public Attributes

long SeatNum
Check's seat number.

Detailed Description

Structure that defines a voucher information by seat.

ResPosAPI_SvcCharge Struct Reference

Structure used to represent a service charge in Transaction Services.

Public Attributes

- long SvcChgObjectNum
  SvcChgObjectNum
  Maps to the service charge object number in the micros database
- BSTR SvcChgAmountOrPercent
  SvcChgAmountOrPercent
  Pass an amount or a percentage
- BSTR SvcChgReference
  SvcChgReference
  Optional reference entry

Detailed Description

Structure used to represent a service charge in the Transaction Services. These fields are all mapped to the micros database query SELECT * FROM "micros"."dsvc_def" where type = 'S'

ResPosAPI_TmedDetailItem Struct Reference

Tender Media Item

Public Attributes

- long TmedObjectNum
  Pass the Tender Media Object number.
- BSTR TmedPartialPayment
  Partial payment amount - for paid in full leave this empty/null.
- BSTR TmedReference
  Tender Media reference Information.
- ResPosAPI_CreditCard Struct Reference TmedCreditCard
  Credit Card Information.

Detailed Description

Tender Media Item

PMS interface is not supported.
Tender Media reference can support more than one reference. The API will take the BSTR passed and assign reference entries for each group of 20 characters it finds. Three reference entries would have a length of 60.

ResPosAPI_TmedDetailItemEx Struct Reference

Tender Media Item Ex Version

Public Attributes

- long TmedObjectNum
  Pass the Tender Media Object number.
- BSTR TmedPartialPayment
  Partial payment amount - for paid in full leave this empty/null.
- BSTR TmedReference
  Tender Media reference Information.
- ResPosAPI_EPayment Struct Reference TmedEPayment
  Electronic Payment Information.

Detailed Description

Tender Media Item Ex Version

The only Tender Media items that will be accepted are service total types and payment types. Tender Options that may be supported:

- **Exempt Auto Service Charge Post To Gross receipts**
- **Credit Authorization, Debit, and Gift Card Payment**

Tender Media reference needs to be space-filled up to 20 characters. Tender Media reference can support more than one reference. The API will take the BSTR passed and assign reference entries for each group of 20 characters it finds. Three reference entries would have a length of 60.

ResPosAPI_TotalsResponse Struct Reference

Totals Response Structure

Public Attributes

- BSTR TotalsSubTotal
  Sub Total
- BSTR TotalsTaxTotals
  Tax total
- BSTR TotalsOtherTotals
  Other Service Charge Total, Same total as the RVC Discount Service Other Service Charge Name
- BSTR TotalsAutoSvcChgTotals
  Auto Service charge Totals
- BSTR TotalsTotalDue
  Total Due

Detailed Description

Totals Response Structure

Simply the amount owed by the user

ResPrintAPI_PrintJob Struct Reference

Structure used to represent a print job

Public Member Functions

SAFEARRAY

(ResPrintAPI_PrintLine) PrintLines
Array of Print lines.

Public Attributes

- long PrimaryPrinterObjectNum
  Primary Printer
- long PrimaryBackupPrinterObjectNum
  Primary Backup Printer
- long SecondBackupPrinterObjectNum
  Secondary Backup Printer
- long StatusID
  Optional Status Value

Detailed Description

Structure used to represent a print job.

ResPrintAPI_PrintJobStatus Struct Reference

Structure that defines the print job status

Public Attributes

- ResPrintAPI_Status Status
  Enumerated status value.
- BSTR StatusMsg
  Print Job status Message.
- BSTR SystemStatusMsg
  System Status Message. Similar to Ops Windowed status messages on failed print job, door open, printer jammed...

Detailed Description

Structure that defines the print job status.
In addition to containing a status value the structure also contains localized strings that describe the current status of the print job and the system.
It is very important to understand what this structure returns to the caller. The printing architecture is broken into two parts:

1. **Local manager**
2. **Remote manager**

The local manager is responsible for delivering your print job to the correct node for printing. The Status element of this structure will report on the success or failure of the local manager to send your print job to the remote node. The StatusMsg element is a textual representation of the enumerated value. It is important to realize that a successful send of your print job does not guarantee your job will print.
The Remote manager is responsible for printing your job and it's the process that talks directly to the printers. The remote manager is commonly known to RES installers and users as Pcontrol. The SystemStatusMsg is what the remote manager will return when some error happens when a print job is sent to a device. This message is not linked in any way with any specific print job, it is simply a system alert that some job failed and that the printer needs attention.
There is currently no way to track a specific print job from start to finish.

4 API Return Codes

POS_PRINT_API

Table 4-1 - POS_PRINT_API

MessageId	MessageText	DWORD
E_PRINT_SVC_DISABLED	Print Services are Disabled	0x87020001L
E_PRIM_PRINTER_NOT_FOUND	No Primary Printer was found	0x87020002L

POS_TRANS_API

Table 4-2 - POS_TRANS_API

MessageId	MessageText	DWORD
E_POSAPI_INTERNAL_ERROR	Internal Error	0xC7010000L
E_POSAPI_INVALID_TEXT_FILE	Unable to load text files	0xC7010001L
E_POSAPI_INVALID_RES_INSTALL	RES POS API is not compatible with the installed version of RES	0xC7010002L
E_POSAPI_BAD_LEVEL	The menu level passed to the function is out of range	0xC7010004L
E_POSAPI_ITEM_NOT_FOUND	Item passed was not found in the database	0xC7010005L
E_POSAPI_WEIGHT_PARAM_MISSING	Menu Item Weight Missing from call	0xC7010006L
E_POSAPI_INVALID_MAJ_FAM_GRP	Invalid Major or Family Group Programming	0xC7010007L
E_POSAPI_INVALID_SLS_ITMZR	Invalid Sales Itemizer	0xC7010008L
E_POSAPI_WEIGHT_TOO_LARGE	Item weight is too large.	0xC7010009L
E_POSAPI_NO_MULTI_INCL_RATE	Multiple inclusive rates are not supported	0xC701000AL
E_POSAPI_TMED_NOT_FOUND	Tender Media Not found	0xC701000BL
E_POSAPI_MENU_LVL_MISMATCH	The level passed to the function does not match the definition information	0xC701000CL

MessageId	MessageText	DWORD
E_POSAPI_FOREIGN_CURRENCY_MISSING	Tender Requires a foreign currency amount	0xC701000DL
E_POSAPI_UNKNOWN_TMED_TYPE	Tender definition is incomplete, cannot find a type	0xC701000EL
E_POSAPI_NO_VOID_WITH_PERS_CHK_OR_CASHBACK	Cannot void a Cash back or a personal check	0xC701000FL
E_POSAPI_CANNOT_USE_WITH_CLOSED_CHECK	Operation not permitted in Closed check edit mode	0xC7010010L
E_POSAPI_MUST_PROVIDE_PAYMENT_AMOUNT	Payment is missing the amount	0xC7010011L
E_POSAPI_PAYMENT_AMOUNT_MUST_BE_ROUNDED	Payment amount must be rounded based on configuration	0xC7010012L
E_POSAPI_CC_NOT_SUPPORTED	Tender Requires Credit Authorization. Payment Command indicates no electronic payment desired.	0xC7010013L
E_POSAPI_DATABASE_NOT_READY	API Database Not Finished Downloading	0xC7010014L
E_POSAPI_NO_RVC_DEFLT_CASH_TMED	No Default cash Tender media Found	0xC7010015L
E_POSAPI_CANNOT_ADD_REF_TO_DTL	Unable To add reference entry to this type of detail	0xC7010016L
E_POSAPI_EMPLOYEE_NOT_FOUND	Could Not Find the Employee In the Database	0xC7010017L
E_POSAPI_DSVC_NOT_FOUND	Could Not Find the Discount Service Charge In the Database	0xC7010018L
E_POSAPI_DSVC_AMT_NOT_FOUND	Could Not Find the Discount Service Charge In the Database	0xC7010019L
E_POSAPI_NO_ITEM_TO_DISCOUNT	No Menu Item Found In Detail To Add Item Discount	0xC701001AL
E_POSAPI_DISCOUNT_IS_TOO_LARGE	Discount is too large	0xC701001BL
E_POSAPI_DISCOUNT_IS_NOT_ITEM_DISCOUNT	Discount Is not an Item discount	0xC701001CL
E_POSAPI_INVALID_COMBO_MENU_ITEM	This item is not a combo meal menu item, it is not linked to any combo meal as a Combo Meal Menu Item	0xC701001DL
E_POSAPI_INVALID_COMBO_MAIN_ITEM	This item is not a combo meal main item, it was not	0xC701001EL

MessageId	MessageText	DWORD
	found in the main item group	
E_POSAPI_INVALID_COMBO_SIDE_ITEM	This item is not a combo meal side item, it was not found in the appropriate side group	0xC701001FL
E_POSAPI_UNSUPPORTED_TENDER_TYPE	The tender media type passed to the interface is not supported, it must be a payment or service total	0xC7010020L
E_POSAPI_DISCOUNT_IS_NOT_STTL_DISCOUNT	Discount Is not a subtotal discount	0xC7010021L
E_POSAPI_CHECK_OBJECT_HAS_CHECK	You cannot pick up a check when the check object already has a check open	0xC7010022L
E_POSAPI_CHECK_WAITING_FOR_AUTH	The desired check is waiting for authorization	0xC7010023L
E_POSAPI_BAD_CHK_SEQ	The desired check sequence is not valid	0xC7010024L
E_POSAPI_RVC_NOT_FOUND	The desired revenue center is not valid	0xC7010025L
E_POSAPI_CLIENT_NOT_LICENSED	The RES API client is not licensed.	0xC7010026L
E_POSAPI_CLIENT_IN_DEMO_MODE	Call succeeded, however, the RES API client is in demo mode.	0x87010027L
E_POSAPI_CLIENT_IN_GRACE_PERIOD	Call succeeded, however, the RES API client licensing is in grace period.	0x87010028L
E_POSAPI_TABLE_OBJ_NOT_FOUND	The table object passed was not found in the revenue center.	0xC7010029L
E_POSAPI_API_EMPL_NOT_FOUND	The employee passed was not an API employee.	0xC701002AL
E_POSAPI_EMPLOYEE_CLASS_NOT_FOUND	The employee class could not be found in the emp status table.	0xC701002BL
E_POSAPI_DELAYED_ORDER_WRONG_BUSINESS_DATE	The delayed order must be submitted on the current business date.	0xC701002CL
E_POSAPI_DELAYED_ORDER_TIME_HAS_PASSED	The delayed order submitted is not in the future.	0xC701002DL

MessageId	MessageText	DWORD
E_POSAPI_SUCCESS_CANCELLED	The operation was cancelled.	0x0701002DL
E_POSAPI_ERROR_BAD_ORDER_TYPE_PARAM	The order type is either out of range or is not set to be active.	0xC701002EL
E_POSAPI_TENDER_NOT_SVC_TTL	The tender media type passed to the interface is not supported, it must be a service total	0xC701002FL
E_POSAPI_SYSTEM_NOT_PCI_COMPLIANT	The system configuration is not PCI compliant	0xC7010030L
E_POSAPI_FUTURE_ORDER_CONVERSION_NOT_ALLOWED	The future order must be declared when the order is first created and cannot be changed once set.	0xC7010031L
E_POSAPI_FUTURE_ORDER_DETAIL_NOT_ALLOWED	The future order may not have certain detail added, including payments and credit authorizations.	0xC7010032L
E_POSAPI_DEFAULT_EXCEPTION	Unexpected condition detected.	0xC70103E9L
E_POSAPI_DB_INTERFACE_EXCEPTION	Invalid database object detected.	0xC70103EAL
E_POSAPI_GUEST_CHECK_EXCEPTION	Invalid guest check structure detected.	0xC70103EBL
E_POSAPI_TOTAL_POST_EXCEPTION	Failed in totals posting routine.	0xC70103ECL
E_POSAPI_TAX_CALC_EXCEPTION	Failed in tax calculation routine.	0xC70103EDL
E_POSAPI_DETAIL_READ_FAILURE	Number of detail items read from the database does not match the check total.	0xC70103EEL
E_POSAPI_CA_AMT_TOO_LARGE	Credit Auth Amount Exceeds Amount Due.	0xC70103EFL
E_POSAPI_CA_ZERO_NOT_ALLOWED	Zero Credit Auth Amount Not Allowed.	0xC70103F0L
E_POSAPI_CA_RESPONSE_UNEXPECTED	CA Response Unexpected Error See Log for error number.	0xC70103F1L
E_POSAPI_CA_RESPONSE_UNKNOWN	CA Response Undefined Error.	0xC70103F2L
E_POSAPI_CA_RESPONSE_ERROR	Error Waiting For Credit Auth Response.	0xC70103F3L
E_POSAPI_CA_UNSUPPORTED_RESP	Unsupported Response from CA Driver.	0xC70103F4L

MessageId	MessageText	DWORD
E_POSAPI_CA_TENDER_NOT_FOUND	Credit Authorization Tender Not Found.	0xC70103F5L
E_POSAPI_CA_RPC_ERR	Error Communicating To Credit Card Server.	0xC70103F6L
E_POSAPI_CA_NOT_INIT	Credit Authorization Not Enabled.	0xC70103F7L
E_POSAPI_ENCRYPTION_INIT_ERROR	Error unable to initialize encryption support services	0xC70103F8L
E_POSAPI_WRONG_NUMBER_OF_CA_DTL	Error cannot add wrong number of Credit auth detail	0xC70103F9L
E_POSAPI_CANNOT_PROCESS_CA_CONDITIONAL_AUTH	The authorization is asking for conditional approval, the API cannot process the conditional approval	0xC70103FAL
E_POSAPI_CANNOT_PROCESS_CA_DECLINED	This credit authorization has been declined	0xC70103FBL
E_POSAPI_CANNOT_PROCESS_CA_DRIVER_ERROR	This credit authorization has been set to an error state by the ca driver	0xC70103FCL
E_POSAPI_CA_MAN_AUTH_REQD	This credit authorization can only be entered by manual authorization at this time	0xC70103FDL
E_POSAPI_CANNOT_PROCESS_CA_PROMPT	The driver is asking for additional user information, the API cannot process the drivers prompt	0xC70103FEL
E_POSAPI_NO_CHARGE_TIP_LINK_FOUND	The tender media charge tip link cannot be found.	0xC70103FFL
E_POSAPI_CA_INVALID_CARD_NUMBER	The credit card passed is not formatted correctly	0xC7010400L
E_POSAPI_CA_INVALID_CARD_EXP_DATE	The credit card expiration date is invalid	0xC7010401L
E_POSAPI_CA_INVALID_CARD_MOD_CHECK	The credit card failed the mod 10 check	0xC7010402L
E_POSAPI_CA_ERROR_REPLY	The credit card subsystem returned an error in the reply structure.	0xC7010403L
E_POSAPI_INTERFACE_NOT_FOUND	The interface name could not be found in the list of interfaces.	0xC7010404L
E_POSAPI_INVALID_DEVICE_NUMBER	The device number is invalid.	0xC7010405L

MessageId	MessageText	DWORD
E_POSAPI_INTERNAL_BUFFER_TOO_SMALL	Internal error.	0xC7010406L
E_POSAPI_REQUEST_TIMEOUT	Timed out waiting for an external response.	0xC7010407L
E_POSAPI_GIFT_CARD_TENDER_REQUIRES_CREDIT_AUTH	The Gift Card Tender is programmed to require credit authorization.	0xC7010408L
E_POSAPI_RESPONSE_MISMATCH	The gift card response does not match the transmitted request.	0xC7010409L
E_POSAPI_GIFT_CARD_VERSION_MISMATCH	Gift Card version mismatch.	0xC701040AL
E_POSAPI_GIFT_CARD_NOT_APPROVED	Gift Card not approved.	0xC701040BL
E_POSAPI_GIFT_CARD_ITEM_NOT_A_TENDER	The gift card redemption item is not a tender.	0xC701040CL
E_POSAPI_UNKNOWN_IFS_ERROR	Unknown interface error.	0xC701040DL
E_POSAPI_IFS_INVALID_CONFIG	Invalid interface configuration.	0xC701040EL
E_POSAPI_IFS_SERVER_NAME_SIZE_ERROR	Interface server name size error.	0xC701040FL
E_POSAPI_IFS_NO_TX_DATA	No interface data to transmit.	0xC7010410L
E_POSAPI_IFS_DATA_SIZE_ERROR	Interface data size error.	0xC7010411L
E_POSAPI_IFS_BIND_ERROR	Interface binding error.	0xC7010412L
E_POSAPI_IFS_ID_SIZE_ERROR	Interface id size error.	0xC7010413L
E_POSAPI_IFS_NOT_REGISTERED	Interface not registered.	0xC7010414L
E_POSAPI_IFS_INVALID_RX_BUFFER	Invalid interface receive buffer.	0xC7010415L
E_POSAPI_IFS_INTERFACE_NOT_ACTIVE	Interface not active.	0xC7010416L
E_POSAPI_IFS_SERVER_NOT_ACTIVE	Interface server not active.	0xC7010417L
E_POSAPI_IFS_RESOURCE_ERROR	Interface resource error.	0xC7010418L
E_POSAPI_IFS_OS_RESOURCE_ERROR	Interface OS resource error.	0xC7010419L
E_POSAPI_IFS_INTERNAL_ERROR	Interface internal error.	0xC701041AL
E_POSAPI_IFS_RECEIVE_TIMEOUT	Interface receive timeout.	0xC701041BL
E_POSAPI_IFS_RPC_ERROR	Interface RPC error.	0xC701041CL
E_POSAPI_IFS_NO_INTERFACE_NAME	No interface name.	0xC701041DL
E_POSAPI_IFS_INTERFACE_NAME_SIZE_ERROR	Interface name size error.	0xC701041EL
E_POSAPI_NO_CHECK_INFO_LINES	No check information lines programmed.	0xC701041FL

MessageId	MessageText	DWORD
E_POSAPI_CHECK_INFO_LINES_FULL	No free check information lines.	0xC7010420L
E_POSAPI_REQUEST_ERROR	Request rejected.	0xC7010421L
E_POSAPI_NOT_LICENSED	Stored Value Card server not licensed.	0xC7010422L
E_POSAPI_CASH_BACK_PASSED_NOT_DEBIT_TMED	A cash back cannot be entered for a non-debit tender media record.	0xC7010423L
E_POSAPI_SVC_SERVER_NOT_FOUND	The Stored Value Card server could not be contacted.	0xC7010424L
E_POSAPI_GIFT_CARD_MISSING_TRACE_ID	The Stored Value Card transaction is missing its trace identifier.	0xC7010425L
E_POSAPI_CANNOT_REPRINT_BC_PREV_VOUCHER_NOT_FOUND	Cannot reprint a voucher for a credit card that is not found on the check.	0xC7010426L
E_POSAPI_CANNOT_FIND_DEBIT_SERVICE_CHARGE	Cannot find the debit service charge, cashback cannot be added to the check.	0xC7010427L
E_POSAPI_GIFT_CARD_TENDER_DOES_NOT_ALLOW_TIPS	Stored Value Card Tender is not programmed for tips.	0xC7010428L
E_POSAPI_GIFT_CARD_DECLINED	Gift Card redemption declined.	0xC7010429L
E_POSAPI_CUSTOM_1	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701042AL
E_POSAPI_CUSTOM_2	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701042BL
E_POSAPI_CUSTOM_3	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701042CL
E_POSAPI_CUSTOM_4	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701042DL
E_POSAPI_CUSTOM_5	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701042EL
E_POSAPI_CUSTOM_6	Custom error: Use IsupportErrorInfo interface	0xC701042FL

MessageId	MessageText	DWORD
	to access the error message.	
E_POSAPI_CUSTOM_7	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010430L
E_POSAPI_CUSTOM_8	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010431L
E_POSAPI_CUSTOM_9	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010432L
E_POSAPI_CUSTOM_10	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010433L
E_POSAPI_CUSTOM_11	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010434L
E_POSAPI_CUSTOM_12	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010435L
E_POSAPI_CUSTOM_13	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010436L
E_POSAPI_CUSTOM_14	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010437L
E_POSAPI_CUSTOM_15	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010438L
E_POSAPI_CUSTOM_16	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010439L
E_POSAPI_CUSTOM_17	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701043AL
E_POSAPI_CUSTOM_18	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701043BL

MessageId	MessageText	DWORD
E_POSAPI_CUSTOM_19	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701043CL
E_POSAPI_CUSTOM_20	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701043DL
E_POSAPI_CUSTOM_21	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701043EL
E_POSAPI_CUSTOM_22	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701043FL
E_POSAPI_CUSTOM_23	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010440L
E_POSAPI_CUSTOM_24	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010441L
E_POSAPI_CUSTOM_25	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010442L
E_POSAPI_CUSTOM_26	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010443L
E_POSAPI_CUSTOM_27	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010444L
E_POSAPI_CUSTOM_28	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010445L
E_POSAPI_CUSTOM_29	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010446L
E_POSAPI_CUSTOM_30	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010447L
E_POSAPI_CUSTOM_31	Custom error: Use IsupportErrorInfo interface	0xC7010448L

MessageId	MessageText	DWORD
	to access the error message.	
E_POSAPI_CUSTOM_32	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010449L
E_POSAPI_CUSTOM_33	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701044AL
E_POSAPI_CUSTOM_34	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701044BL
E_POSAPI_CUSTOM_35	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701044CL
E_POSAPI_CUSTOM_36	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701044DL
E_POSAPI_CUSTOM_37	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701044EL
E_POSAPI_CUSTOM_38	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701044FL
E_POSAPI_CUSTOM_39	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010450L
E_POSAPI_CUSTOM_40	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010451L
E_POSAPI_CUSTOM_41	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010452L
E_POSAPI_CUSTOM_42	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010453L
E_POSAPI_CUSTOM_43	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010454L

MessageId	MessageText	DWORD
E_POSAPI_CUSTOM_44	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010455L
E_POSAPI_CUSTOM_45	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010456L
E_POSAPI_CUSTOM_46	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010457L
E_POSAPI_CUSTOM_47	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010458L
E_POSAPI_CUSTOM_48	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC7010459L
E_POSAPI_CUSTOM_49	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701045AL
E_POSAPI_CUSTOM_50	Custom error: Use IsupportErrorInfo interface to access the error message.	0xC701045BL
E_POSAPI_GIFT_CARD_ERROR	Error submitting the Gift Card request	0xC701045CL
E_POSAPI_RETRANSMIT	Retransmit the Gift Card request.	0xC701045DL
E_POSAPI_DEBIT_OVERTENDER_NOT_ALLOWED	The sum of the payment, tip, and cashback exceed the amount due, cannot apply payment.	0xC701045EL
E_POSAPI_CLIENT_CONFIG_NOT_FOUND	The API client configuration record could not be found.	0xC701045FL
E_POSAPI_EMPL_DOES_NOT_OWN_CHECK	The employee passed to the API is not the Check employee for this check.	0xC7010460L
E_POSAPI_PMS_POSTING_DENIED	The PMS charge has been declined.	0xC7010461L
E_POSAPI_PMS_ACCOUNT_SELECTION_REQUIRED	PMS Account selection is required.	0x87010462L
E_POSAPI_ISL_PMS_NOT_SUPPORTED	Enhanced PMS posting is not supported.	0xC7010463L

MessageId	MessageText	DWORD
E_POSAPI_INTFC_SW_NOT_LICENSED	Interface Software Not Licensed.	0xC7010464L
E_POSAPI_REF_NUM_REQUIRED	Entry Required.	0xC7010465L
E_POSAPI_NO_RESPONSE_FROM_PMS	No response from PMS.	0xC7010466L