MS Govern A/R & Cash Collection Service API

Release 6.0/6.1

Last Revision Update: 08/28/2017

Account Receivable Service API
Document History

Issue Date	Version	Details	Author
March 7, 2013	1.0 (Draft)	API description of the Accounts Receivable & Cash Collection service	Benoît Lauzon
March 11, 2013	1.1	Initial version reviewed for distribution to Invoice Cloud	Fabrice Olivier
March 21, 2013	1.1	*Add userID for method* ***AR_ExecutePaymentReversalByReceiptNo***	Zahir Kaci
March 25, 2013	1.2	Revision of the Mock Up Section	Benoît Lauzon
April 17th, 2013	1.2	Add EffectiveDate property to the class ***DC_AR_PaymentSummary***	Zahir Kaci
August 28th, 2017	6.1.1706	Add **AR_SetAccountPaperlessIndicatorOFF** Method Add **AR_ExecutePaymentKeyBank** Method and class **DC_KeyBankResult**	Zahir Kaci

Summary Table

Introduction

This document describes the MSGovern Accounts Receivable and Cash Collection System Service API.

NOTE: Windows Communication Foundation (WCF) is an API within the .NET Framework. Although the Govern A/R & CC WCF Web Service supports JavaScript Object Notation (JSON) serialization format to WCF, *MS Govern* does not provide developmental technical support for this type of implementation.

Disclaimer

MS Govern Software applications are developed using Microsoft's .NET Technology. Applications are intended for use within the .NET Framework 4.0.
MS Governs' API Documentation is provided with the understanding that implementation of described methods and contracts will be with .NET Development tools and supported programming languages. Users should note that with the exception of MS Governs' Methods or Contracts that are implemented with Visual Basic .NET and C#, MS Govern Software does not provide development guidance or support.

MS Govern has taken due care in preparing this manual. However, nothing contained herein modifies or alters in any way the standard terms and conditions of the purchase, lease, or license agreement by which the product was acquired, nor increases in any way the liability of MS Govern to the customer.

I. Service contract

The Accounts Receivable service implements the following methods:

1) AR_GetBalancesByInvoice

```
Function AR_GetBalancesByInvoice(
      ByVal objAuthentification As DC_Authentification,
      ByVal strInvoiceNumber As String
      ByVal dateBalanceAsOf As Date
      ) As DC_AR_BalanceByInstallment
```

This method will return the balance as of date including penalties and interest for a given invoice. MS Govern do not have the concept of an Invoice number. To simulate this, the Invoice number is [AR_ID].[INSTALLMENT]

Return

- DC_AR_BalanceByInstallment

Exception

- FE_BaseException
- DC_FE_NonExistingUser

2) AR_GetBalancesByAccountId

```
Function AR_GetBalancesByAccount(
      ByVal objAuthentification As DC_Authentification,
      ByVal intAccountNumber As Integer,
      ByVal booIncludeInvoiceWithoutBalance As Boolean
      ByVal dateBalanceAsOf As Date
      ) As List(Of DC_AR_AccountBalance)
```

This method will return the balance as of date for a given Account. MS Govern do not have the concept of a global account, but use the NA_ID simulate this. We will retrieve all Mailing index records for a given Account (NA_ID), and will create a DC_AR_AccountBalance for each AR records found that has a balance. Each AR record will have a list of balance by installment (DC_AR_BalanceByInstallment).

If booIncludeInvoiceWithoutBalance is set to true, we will include all AR even if the balance has been paid.

**Currently only UB and Real Property Tax Accounts receivable will be returned by this method. Other sub systems will be included in a future release.*

Return

- DC_AR_BalanceByInstallment

Exception

- FE_BaseException
- DC_FE_NonExistingUser

3) AR_SetAccountToPaperless

```
Function AR_SetAccountToPaperless(
      ByVal objAuthentification As DC_Authentification,
      ByVal intAccountNumber As Integer,
      ByVal strEMailAddress As String
      )
```

This method will set the paperless flag for a given account. MS Govern do not have the concept of a global account, but we use the NA_ID (associated to a unique name record) to simulate this. The method will set the paperless flag (NA_MAILING_INDEX.BILL_BY_EMAIL) and the email address (NA_MAILING_INDEX.EMAIL_ADDRESS) for each mailing index record found.

Exception

- FE_BaseException
- DC_FE_NonExistingUser

4) *AR_SetAccountPaperlessIndicatorOFF*

```
 Sub AR_SetAccountPaperlessIndicatorOFF(authentification As DC_Authentification,
                                        accountNumber As Integer)
```

This method will set the paperless flag to Off for a given account. MS Govern do not have the concept of a global account, but we use the NA_ID (associated to a unique name record) to simulate this. The method will set the paperless flag (NA_MAILING_INDEX.BILL_BY_EMAIL) to **False** for each mailing index record found.

Exception

- FE_BaseException
- DC_FE_NonExistingUser

5) AR_ExecutePayment

```
Function AR_ExecutePayment(
      ByVal objAuthentification As DC_Authentification,
      ByVal objArPaymentSummary As DC_AR_PaymentSummary
      ) As Integer
```

This method will create a payment in the MS Govern database.

- The combination of the User Id and the Receipt NO must be unique.
- BatchNo and EntryDate are not required, if left blank, the system will generate these values.

Return

- Payment Unique identifier (CC_ID)

Exception

- FE_BaseException
- DC_FE_NonExistingUser

6) AR_ExecutePaymentReversal

```
Function AR_ExecutePaymentReversal(
      ByVal objAuthentification As DC_Authentification,
      ByVal intCC_ID As Integer
      ) As Integer
```

This method will reverse a payment with the MS Govern unique Payment identifier.
The method will return the Payment Reversal Unique identifier (CC_ID)

Return

- Payment Unique identifier (CC_ID)

Exception

- FE_BaseException
- DC_FE_NonExistingUser

7) AR_ExecutePaymentReversalByReceiptNo

```
Function AR_ExecutePaymentReversalByReceiptNo(
      ByVal objAuthentification As DC_Authentification,
      ByVal strReceiptNo As String,
      ByVal strUserID As String
      ) As Integer
```

This method will reverse a payment; we will retrieve the payment unique identifier with the receipt no and the user id.

The method will return the Payment Reversal Unique identifier (CC_ID)

Return

- Payment Unique identifier (CC_ID)

Exception

- FE_BaseException
- DC_FE_NonExistingUser

8) Key Bank Integration : *AR_ExecutePaymentKeyBank*

```
Public Function AR_ExecutePaymentKeyBank(
                authentification As DC_Authentification,
                 paymentNotificationInfo As String
                ) As DC_KeyBankResult
```

Cuyahoga County will be using Key Bank's web-based CSR Application (Payment Center) to process credit card transactions from cashiering stations (Govern will be used for Cash/Check transactions). Govern will need to provide a Payment Notification Web Service that can be utilized by Key Bank payment processing applications

This method is added to support the Key Bank payment notification and will use key-value pairs as input parameter.

Payment Notifications string (Parameter: paymentNotificationInfo) will be sent from Key Bank per the Payment Notifications Specification in the Appendix below. Govern's P_ID should be utilized as the

Account Number, and the Payment Confirmation Number should be stored in the CC_MASTER.WS_TRANS_ID field.

Parameter	Description	Format	Comments/Valid Values	Govern Mapping
mtype	Type of notification message.	N[2]	**41 = Payment Scheduled** **42 = Payment Edited** **44 = Payment Cancelled**	If 41 - process payment. If 42 - Do not process in Govern, but note in the log that a payment was modified with details of payment modification. If 43 - Locate paymnet and reverse if found.
acctno	The account number the payment is for.	AN[32]	As supplied by Partner	P_ID. Distribute payment across AR records according to County payment distribution rules.
customername	The full name of the customer the payment is for.	AN[40]	As supplied by Partner	CC_MASTER.NOTES with label "Payer name: " Note in the log if this value is different than the LAST_NAME or COMPANY value in the associated name record for the NA_ID passed back for the custid field.
pymntdate	The date that the payment is scheduled for.	C[10]	MM/DD/YYYY	AR_DETAIL.ENTRY_DATE of pmt record. CC:MASTER.ENTRY_DATE
pymntamnt	The amount of the payment to be processed	N[16,2]		AR_DETAIL.AMOUNT of pmt recprd. CC_MASTER.AMOUNT Note: for returns, validate that the amount returned is the same as the original payment amount. If there is a difference, return/reverse the original payment amount, and note the discrepancy in the log.
confno	The confirmation number for the payment to be processed.	AN[10]		CC_MASTER.WS_TRANS_ID
pymntmethod	Payment Fulfillment Method	N[2]	21 = ACH 22 = Paper Check 23 = Visa Network Debit 24 = Pinless Debit/ATM Network 25 = MasterCard Network Debit 26 = American Express Network Credit 27 = Discover Network Credit 28 = Visa Network Credit 29 = MasterCard Network Credit 30 = CASH 31 = Real-Time ACH Network	Corresponding CC_DETAIL.AMOUNT_TYPE
custid	Customer ID associated with the account the payment was made for	AN[32]		NA_ID passed back
txnentrydatetime (**Waiting confirmation**	The date when customer initiates the payment.	????	YYMMDD:HHMMSS	AR_DETAIL.EFFECTIVE_DATE of pmt record.

from Bill for the name)				

This Method will returns a DC_KeyBankResult object which contains :

- The CC_ID of the processed transaction (Payment Created, Payment Reversed or Deleted, Payment Edited)
- A log Message for warnings or Errors.

Example of the Key-value pair Payment notification string sent by Key Bank to our web service:

http://localhost:51156/SV_AccountsReceivableAPI.svc?mtype=42&pkey=1234567890&acctno=139103&customername=Jean Maurice_KK&pymntdate=08/10/2017&pymntamnt=1800.50&feeamt=1.23&confno=A1B2C3D4E9&pymntchnl=05&pymntype=01&dbacctrtno=123456789&dbacctno=A1B2C3&dbaccttype=02&dbacctsubtype=04&txncode=01&pymntmethod=28&agentloginid=cviron&custid=2408224&=txnentrydatetime=08/29/2017

Return

- `DC_KeyBankResult`

Exception

- `FE_BaseException`
- `DC_FE_NonExistingUser`

II. Data contracts

Account Receivable service provides the following data contracts:

1) DC_AR_AccountBalance

```
Public Class DC_AR_AccountBalance
   Public Property AccountNumber As Integer
   Public Property AR_ID As Integer
   Public Property AR_Description As String
   Public Property BillYear as SmallInt
   Public Property BillNumber as String
   Public Property CurrentBalance As Decimal
   Public Property LateCharges As Decimal
   Public Property TotalBalance As Decimal
   Public Property BalanceByInstallment As List(Of DC_AR_BalanceByInstallment)
End Class
```

_* MS Govern does not have the concept of a global account, but we use the NA_ID simulate this._

2) DC_AR_BalanceByInstallment

```
Public Class DC_AR_BalanceByInstallment
   Public Property AR_ID As Integer
   Public Property Installment As String
   Public Property InvoiceNumber As String
   Public Property CurrentBalance As Decimal
   Public Property LateCharges As Decimal
   Public Property TotalBalance As Decimal
End Class
```

_* MS Govern does not have the concept of an Invoice number. To simulate this, the Invoice number is [AR_ID].[INSTALLMENT]_

3) DC_AR_PaymentDetail

```
Public Class DC_AR_PaymentDetail
   Public Property AR_ID As Integer
   Public Property Installment As String
   Public Property InvoiceNumber As String
   Public Property Amount As Decimal
Class
```

*you must provide one of the following:

- _AR_ID_
- _AR_ID and Installment_
- _InvoiceNumber_

4) DC_AR_PaymentSummary

```
Public Class DC_AR_PaymentSummary
   Public Property ReceiptNo As String(12)
   Public Property BatchNo As String(15)
   Public Property EntryDate As Date
   Public Property EffectiveDate As Date
   Public Property PaymentType As EN_AR_PaymentType
   Public Property PaymentDetail() As List(Of DC_AR_PaymentDetail)
End Class
```

**BatchNo and EntryDate are not required, if left blank, the system will generate the values.*

5) DC_Authentification

```
Public Class DC_Authentification
   Public Property UserID As String
   Public Property UserPassword As String
End Class
```

6) DC_KeyBankResult

```
 Public Class DC_KeyBankResult

   Public Property CC_ID As Integer
   Public Property LogMessage As String

End Class
```

This class is only used for KeyBank integration to return the result of the method **AR_ExecutePaymentKeyBank.**

III. Enumerators

Account Receivable service uses the following enumerator:

1) EN_AR_PaymentType

```
Public Enum EN_AR_PaymentType

    None = 0
    InvoiceCloud = 1

End Enum
```

IV. Mockup

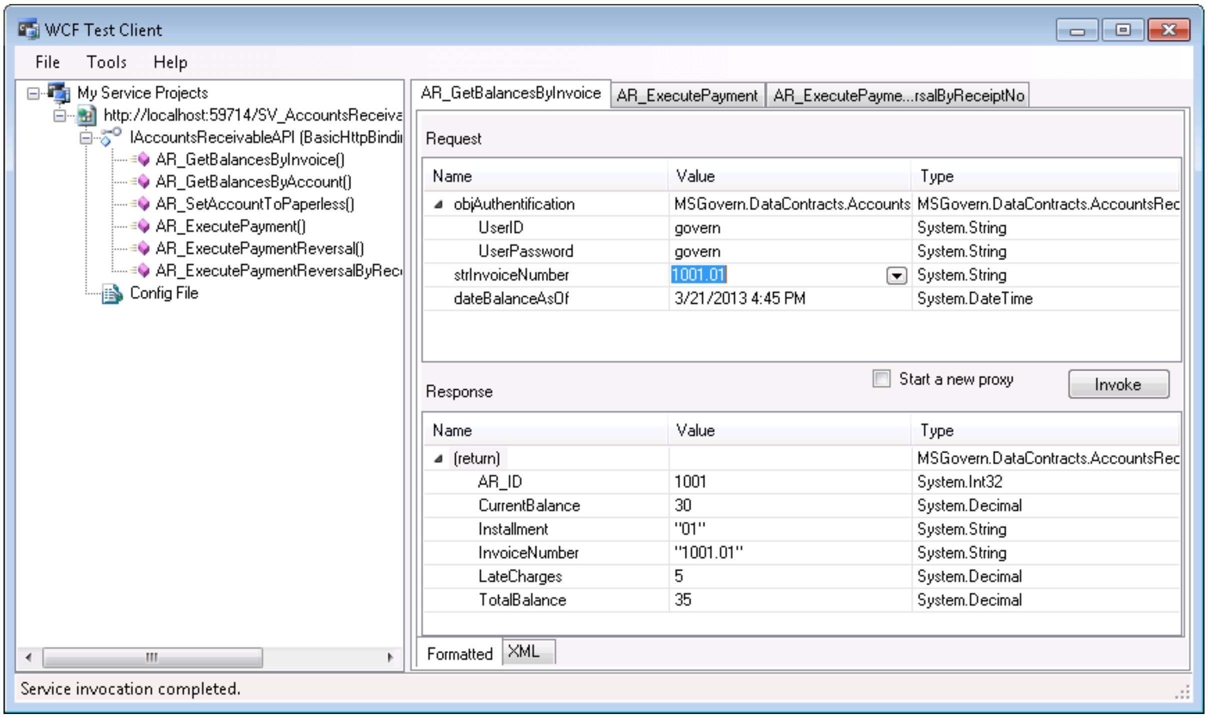
A special compilation allows the service to be consumed without any interaction with the Govern Database and business logic. When using the mockup version of the service, we can test the integration with a third party without affecting the Govern Database.

In the mockup version, if a method request user authentication, the only user valid is “govern” with the password “govern”.

An XML file (MK_AccountsReceivableDB.xml) will be created in the service directory if the file doesn’t exist. This file contains dummy data that the Mockup web service will use to simulate database access (read/write). You can modify the XML file to add more accounts to meet your testing scenarios.

Here are some samples using the mockup methods:

AR_GetBalanceByInvoice


WCF Test Client
File
Tools
Help
My Service Projects
http://localhost:59714/SV_AccountsReceiva
IAccountsReceivableAPI (BasicHttpBindi
AR_GetBalancesByInvoice()
AR_GetBalancesByAccount()
AR_SetAccountToPaperless()
AR_ExecutePayment()
AR_ExecutePaymentReversal()
AR_ExecutePaymentReversalByRec
Config File
AR_GetBalancesByInvoice
AR_ExecutePayment
AR_ExecutePayme...rsalByReceiptNo
Start a new proxy
Invoke
Formatted
XML

Request

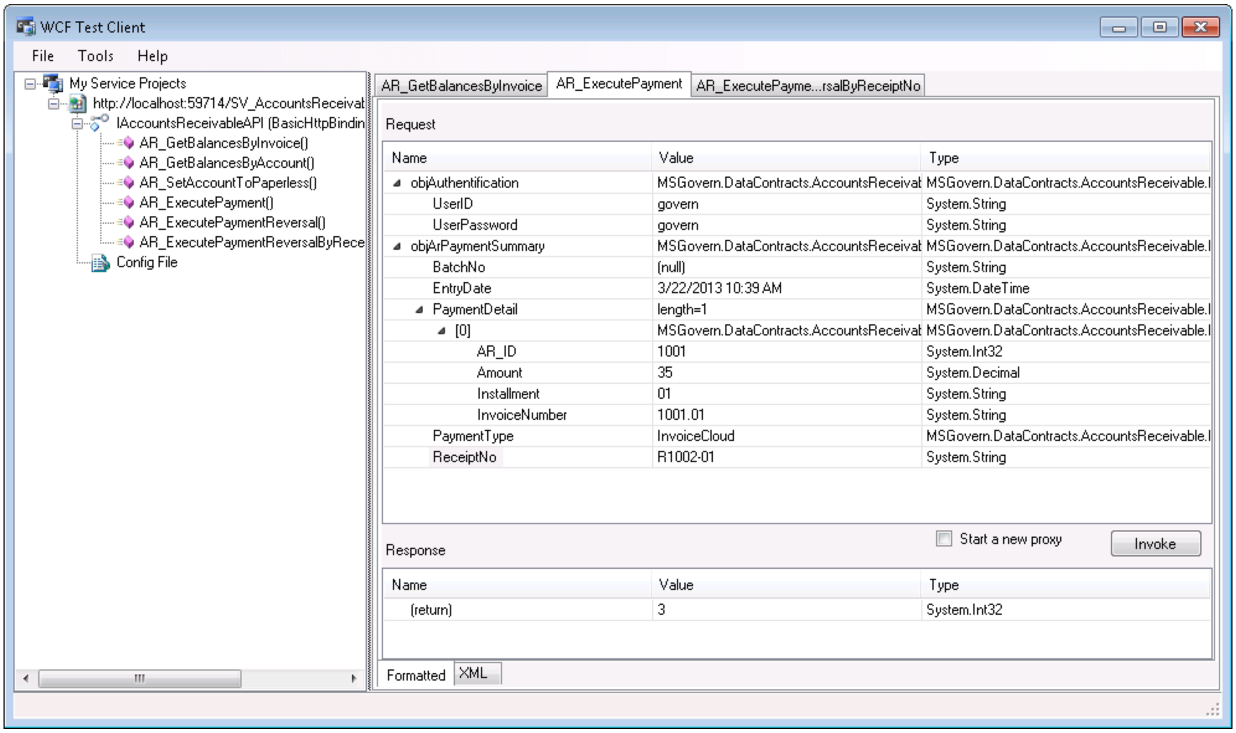
Name	Value	Type
objAuthentification	MSGovern.DataContracts.Accounts	MSGovern.DataContracts.AccountsRec
UserID	govern	System.String
UserPassword	govern	System.String
strInvoiceNumber	1001.01	System.String
dateBalanceAsOf	3/21/2013 4:45 PM	System.DateTime

Response

Name	Value	Type
(return)		MSGovern.DataContracts.AccountsRec
AR_ID	1001	System.Int32
CurrentBalance	30	System.Decimal
Installment	"01"	System.String
InvoiceNumber	"1001.01"	System.String
LateCharges	5	System.Decimal
TotalBalance	35	System.Decimal

Service invocation completed.

AR_ExecutePayment


WCF Test Client
File
Tools
Help
My Service Projects
http://localhost:59714/SV_AccountsReceivat
IAccountsReceivableAPI (BasicHttpBindin
AR_GetBalancesByInvoice()
AR_GetBalancesByAccount()
AR_SetAccountToPaperless()
AR_ExecutePayment()
AR_ExecutePaymentReversal()
AR_ExecutePaymentReversalByRece
Config File
AR_GetBalancesByInvoice
AR_ExecutePayment
AR_ExecutePayme...rsalByReceiptNo
Start a new proxy
Invoke
Formatted
XML

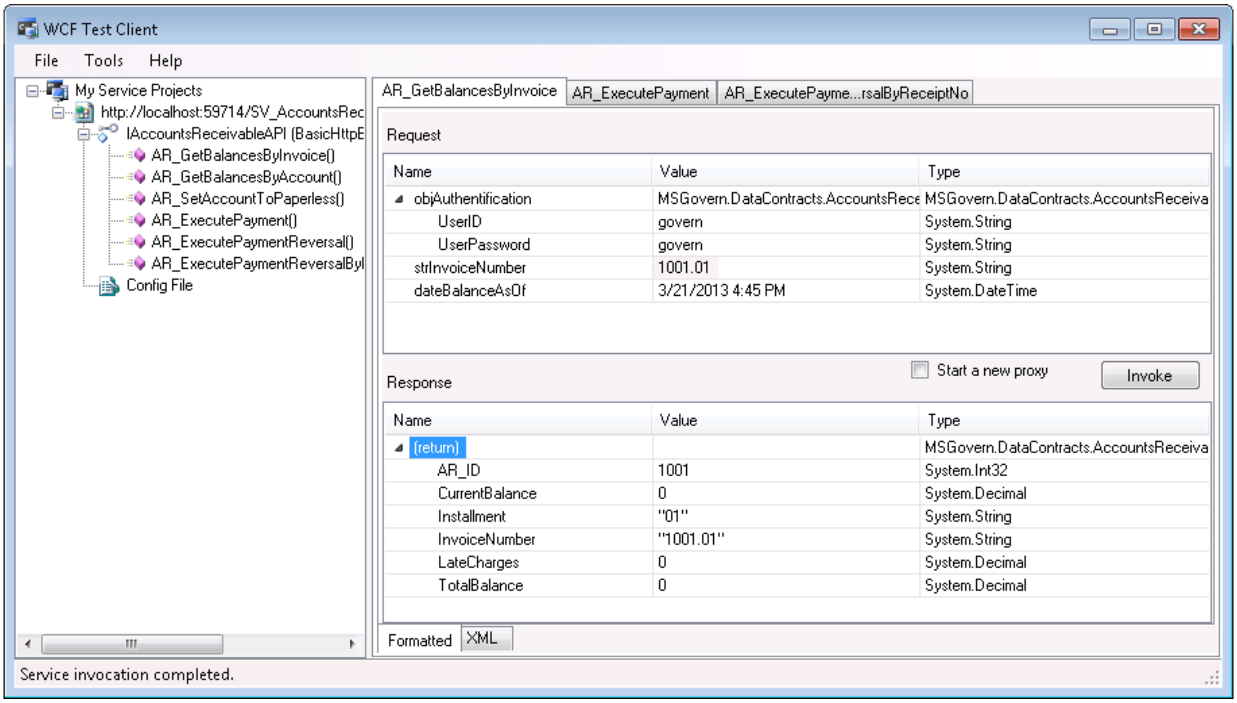
Request

Name	Value	Type
◢ objAuthentification	MSGovern.DataContracts.AccountsReceivat	MSGovern.DataContracts.AccountsReceivable.I
UserID	govern	System.String
UserPassword	govern	System.String
◢ objArPaymentSummary	MSGovern.DataContracts.AccountsReceivat	MSGovern.DataContracts.AccountsReceivable.I
BatchNo	(null)	System.String
EntryDate	3/22/2013 10:39 AM	System.DateTime
◢ PaymentDetail	length=1	MSGovern.DataContracts.AccountsReceivable.I
◢ [0]	MSGovern.DataContracts.AccountsReceivat	MSGovern.DataContracts.AccountsReceivable.I
AR_ID	1001	System.Int32
Amount	35	System.Decimal
Installment	01	System.String
InvoiceNumber	1001.01	System.String
PaymentType	InvoiceCloud	MSGovern.DataContracts.AccountsReceivable.I
ReceiptNo	R1002-01	System.String

Response

Name	Value	Type
(return)	3	System.Int32

AR_GetBalanceByInvoice (after executing the previous payment we will get a new zero balance)


WCF Test Client
File
Tools
Help
My Service Projects
http://localhost:59714/SV_AccountsRec
IAccountsReceivableAPI (BasicHttpE
AR_GetBalancesByInvoice()
AR_GetBalancesByAccount()
AR_SetAccountToPaperless()
AR_ExecutePayment()
AR_ExecutePaymentReversal()
AR_ExecutePaymentReversalByI
Config File
AR_GetBalancesByInvoice
AR_ExecutePayment
AR_ExecutePayme...rsalByReceiptNo
Start a new proxy
Invoke
Formatted
XML

Request

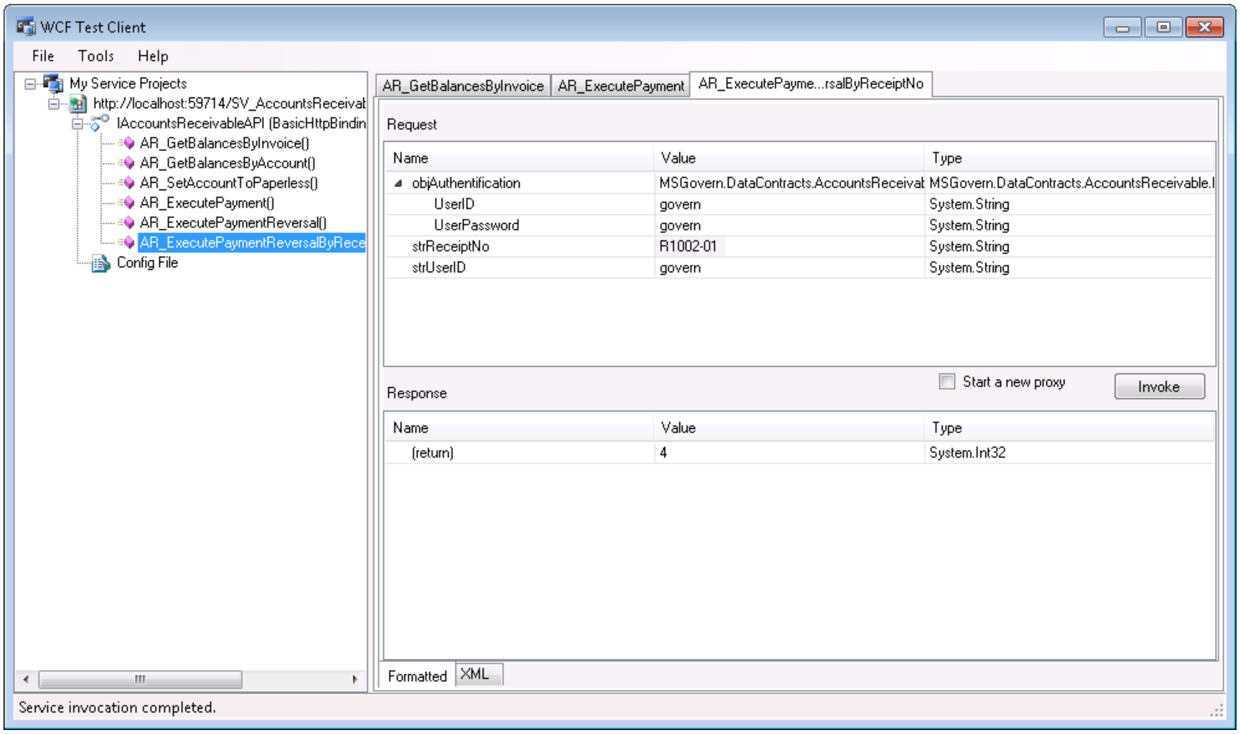
Name	Value	Type
◢ objAuthentification	MSGovern.DataContracts.AccountsRece	MSGovern.DataContracts.AccountsReceiva
UserID	govern	System.String
UserPassword	govern	System.String
strInvoiceNumber	1001.01	System.String
dateBalanceAsOf	3/21/2013 4:45 PM	System.DateTime

Response

Name	Value	Type
◢ (return)		MSGovern.DataContracts.AccountsReceiva
AR_ID	1001	System.Int32
CurrentBalance	0	System.Decimal
Installment	"01"	System.String
InvoiceNumber	"1001.01"	System.String
LateCharges	0	System.Decimal
TotalBalance	0	System.Decimal

Service invocation completed.

AR_ExecutePaymentReversalByReceiptNo


WCF Test Client
File
Tools
Help
My Service Projects
http://localhost:59714/SV_AccountsReceivat
IAccountsReceivableAPI (BasicHttpBindin
AR_GetBalancesByInvoice()
AR_GetBalancesByAccount()
AR_SetAccountToPaperless()
AR_ExecutePayment()
AR_ExecutePaymentReversal()
AR_ExecutePaymentReversalByRece
Config File
AR_GetBalancesByInvoice
AR_ExecutePayment
AR_ExecutePayme...rsalByReceiptNo
Start a new proxy
Invoke
Formatted
XML

Request

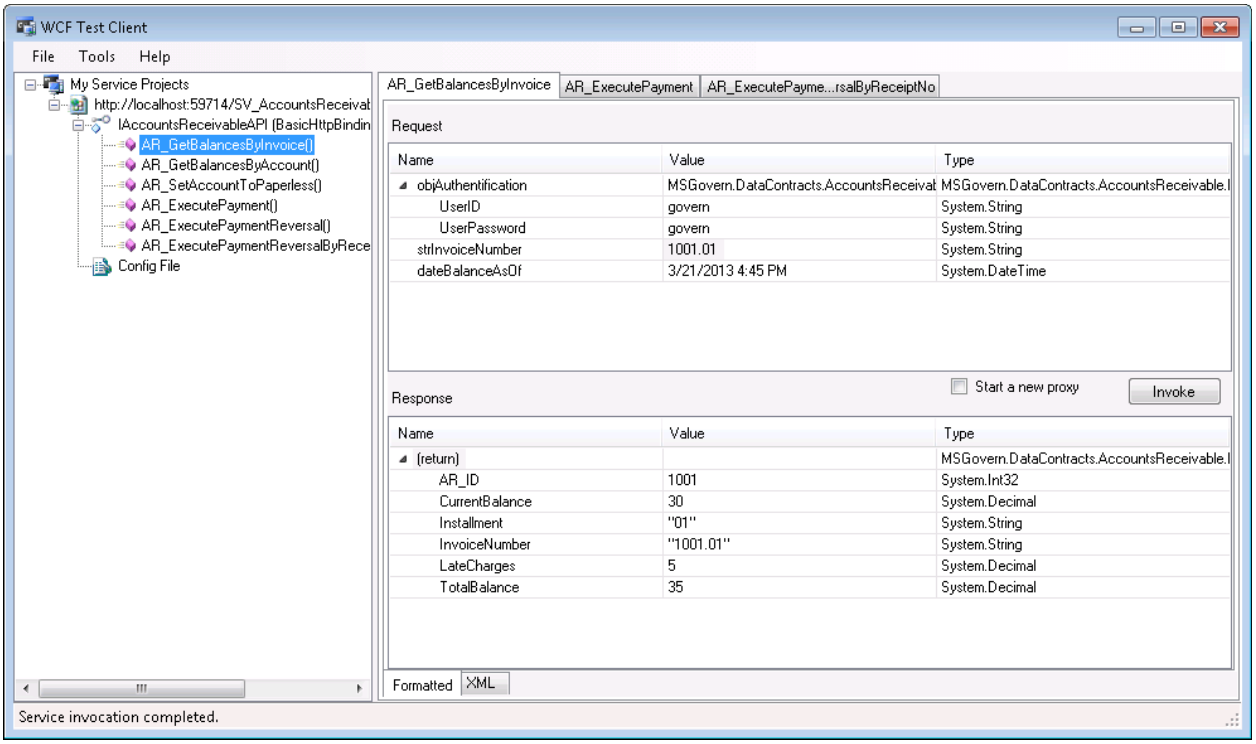
Name	Value	Type
◢ objAuthentification	MSGovern.DataContracts.AccountsReceivat	MSGovern.DataContracts.AccountsReceivable.I
UserID	govern	System.String
UserPassword	govern	System.String
strReceiptNo	R1002-01	System.String
strUserID	govern	System.String

Response

Name	Value	Type
(return)	4	System.Int32

Service invocation completed.

AR_GetBalanceByInvoice (after executing the previous payment reversal we will get a new balance)


WCF Test Client
File
Tools
Help
My Service Projects
http://localhost:59714/SV_AccountsReceivat
IAccountsReceivableAPI (BasicHttpBindin
AR_GetBalancesByInvoice()
AR_GetBalancesByAccount()
AR_SetAccountToPaperless()
AR_ExecutePayment()
AR_ExecutePaymentReversal()
AR_ExecutePaymentReversalByRece
Config File
AR_GetBalancesByInvoice
AR_ExecutePayment
AR_ExecutePayme...rsalByReceiptNo
Start a new proxy
Invoke
Formatted
XML

Request

Name	Value	Type
◢ objAuthentification	MSGovern.DataContracts.AccountsReceivat	MSGovern.DataContracts.AccountsReceivable.I
UserID	govern	System.String
UserPassword	govern	System.String
strInvoiceNumber	1001.01	System.String
dateBalanceAsOf	3/21/2013 4:45 PM	System.DateTime

Response

Name	Value	Type
◢ (return)		MSGovern.DataContracts.AccountsReceivable.I
AR_ID	1001	System.Int32
CurrentBalance	30	System.Decimal
Installment	"01"	System.String
InvoiceNumber	"1001.01"	System.String
LateCharges	5	System.Decimal
TotalBalance	35	System.Decimal

Service invocation completed.

V. Exceptions

The Account Receivable service uses strongly-typed fault exceptions to allow service consumers to have better exception handling.

The service can throw the following exceptions:

1) DC_FE_BaseException

```
Public Class DC_FE_BaseException
   Public Property DBConnectionKey As String
   Public Property OperationName As String
   Public Property Message As String
   Public Property FaultCode As String
   Public Property OriginalRequest As String
End Class
```

2) DC_FE_NonExistingUser

```
Public Class DC_FE_NonExistingUser Inherits DC_FE_BaseException
End Class
```