REAL ESTATE SOFTWARE

MRI

Version 10.4.3
Integrations Guide

September 2017

Every released version of MRI products gets its own release notes. However, guides are only updated when changes have been made to product features that require changes to documentation. If there is no guide specific to your version, use the most recent corresponding document (which may refer to an earlier version of the software).

Table of Contents

Chapter 1 Getting Started 5
Audience 5
Abbreviations 5

Chapter 2 Setting Up MRI for Integration 6
MRI Environment Prerequisites 6
MRI Web Services Description 8
Getting Started as a SaaS Client 8

Chapter 3 Calling MRI APIs 9
Calling APIs as a Simple HTTP Request 9
Calling APIs as SOAP Services 11
Calling APIs Asynchronously 14
Asynchronous Processing Callback 15

Chapter 4 Building-Block APIs 17
Base APIs 17
Currency Exchange Rate Create Interface (IRESMRIBaseCurrencyExchangeCreate) 17
Period Exchange Rate Create Interface (IRESMRIBasePeriodExchangeCreate) 18
Accounts Payable APIs 20
Allocation Read Interface (IRESMRIAPAllocationRead) 20
Tax Read Interface (IRESMRIAPTaxRead) 21
Vendor Read Interface (IRESMRIAPVendorRead) 23
Vendor Creation Interface (IRESMRIAPVendorCreate) 25
Vendor Update Interface (IRESMRIAPVendorUpdate) 27
Vendor Alternate Address Creation Interface (IRESMRIAPVendorAltAddressCreate) 29
Vendor Alternate Address Update Interface (IRESMRIAPVendorAltAddressUpdate) 31
Invoice Creation Interface (MRIAPInvoiceCreate) 32
Payment Read Interface (MRIAPPaymentRead) 36
Payment Marked Read Interface (IRESMRIAPPaymentMarkedRead) 38
Commercial Management APIs 40
Building Read Interface (IRESMRICMBuildingRead) 40
CM Charges (MRICMCharges) 42
Lease Read Interface (LEAS) 44
Expense Pool Accounts Read Interface (IRESMRICMExpensePoolRead) 46
General Ledger APIs 48
Account Update Interface (IRESMRIGLAccountUpdate) 48
Department Update Interface (IRESMRIGLDepartmentUpdate) 50
Job Code Update Interface (IRESMRIGLJobCodeUpdate) 51
Journal Entry Read Interface (IRESMRIGLJournalRead) 53
Journal Entry Marked Read Interface (IRESMRIGLJournalMarkedRead) 55

Journal Entry Create Interface (IRESMRIGLJournalCreate) 57
GL Summary Read Interface (IRESMRIGLSummaryRead) 60
Budget Read Interface (IRESMRIGLBudgetRead) 61
Budget Update Interface (IRESMRIGLBudgetUpdate) 63
Budget Type Read Interface (IRESMRIGLBudgetTypeRead)..... 65
Budget Type Update Interface (IRESMRIGLBudgetTypeUpdate)..... 66
Ledger and Account Read Interface (IRESMRIGLLedgerAccountRead)68
Entity Read Interface (IRESMRIGLEntityRead) 69
Entity Cash Type Map Read Interface (IRESMRIGLEntityCashTypeMapRead)..... 71
Department Read Interface (IRESMRIGLDepartmentRead) 72
Job Code Read Interface (IRESMRIGLJobCodeRead)..... 74
JobCost APIs..... 76
Interface Chart Interface (IRESMRIJCInterfaceRead) 76
Phase Read Interface (IRESMRIJCPhaseRead) 77
Phase Update Interface (IRESMRIJCPhaseUpdate) 79
Cost Code List Read Interface (IRESMRIJCCostListRead) 80
Cost Code Read Interface (IRESMRIJCCostCodeRead)..... 81
Cost Category Read Interface (IRESMRIJCCostCategoryRead) 83
Residential Management APIs..... 85
Guest Card Creation (MRIRMGuestCardCreate)..... 85
Resident Interface (IRESMRIRMResidentEdit) 86

Chapter 5 Solution Integrations.....89
Utility Billing Integration..... 89
MITS 3.0 90
MITS 2.0 97

Chapter 6 Defining Custom APIs 102
Defined Read and Marked Read Schema..... 102
Generic Update Schema..... 104
Stored Procedure Schema 105
Custom Object Schema..... 105
Stored Procedure Method Usage..... 107
Web Services API Definition Schema 107
API Response 108

Appendix A Error Codes.....110

Getting Started 1

MRI Software offers flexible and configurable products that enable our clients to integrate with complementary, best-of-breed software partners to solve complex business problems. The purpose of this *Integrations Guide* is to deliver knowledge to clients and partners who want to leverage MRI APIs and integration capabilities. Readers will learn how to prepare MRI for integrations, gain a deep understanding of the existing API framework and structure, and identify why errors may occur. With this information, clients can enhance their business processes, drive reporting and intelligence, and create ideal workflows that are specific to the needs of their organization.

Audience

This guide assumes the reader has working knowledge of HTTP and XML. It is intended for the following people:

- Software engineers or integration specialists who are building software they would like to integrate with MRI Property Management System
- MRI system administrators who are providing support for the software engineers who are building integrations

Abbreviations

AP—Accounts Payable

API—Application Programming Interface

CM—Commercial Management

GL—General Ledger

JC—JobCost

MITS—Multifamily Information and Transactions Standards

RM—Residential Management

RUBS—Ratio Utility Billing System

Setting Up MRI for Integration 2

This chapter is intended for MRI system administrators.

MRI Environment Prerequisites

Before you can begin using MRI Web Services APIs, your installation of MRI must meet the following prerequisites:

- **MRIAPIServices Web Service**—The MRIAPIServices Web Service is installed with MRI for the Web. The Web Service contains the APIService.asmx file that defines the Web Service.

  Note

  The installation also includes the deprecated MRIWebServices Web Service.

- **Web Services Users**—One or more Web Services users must be set up in the MRI Security and System Administration Console. For more information on adding Web Services users, refer to the "Adding Web Services Users and Assigning Web Services Roles" section in the *System Administration Guide*. Note that the Web Services user does not have access to user interfaces in MRI for Windows or MRI for the Web.

  Caution!
  Because Web Services users will make use of an available license, you must have an available user license to execute API calls. For example, if you have five user licenses, and five users are logged on, API calls will fail.

The Web Services User role must be assigned access to each of the necessary API functions for that user in the Security and System Administration Console (Figure 2-1).

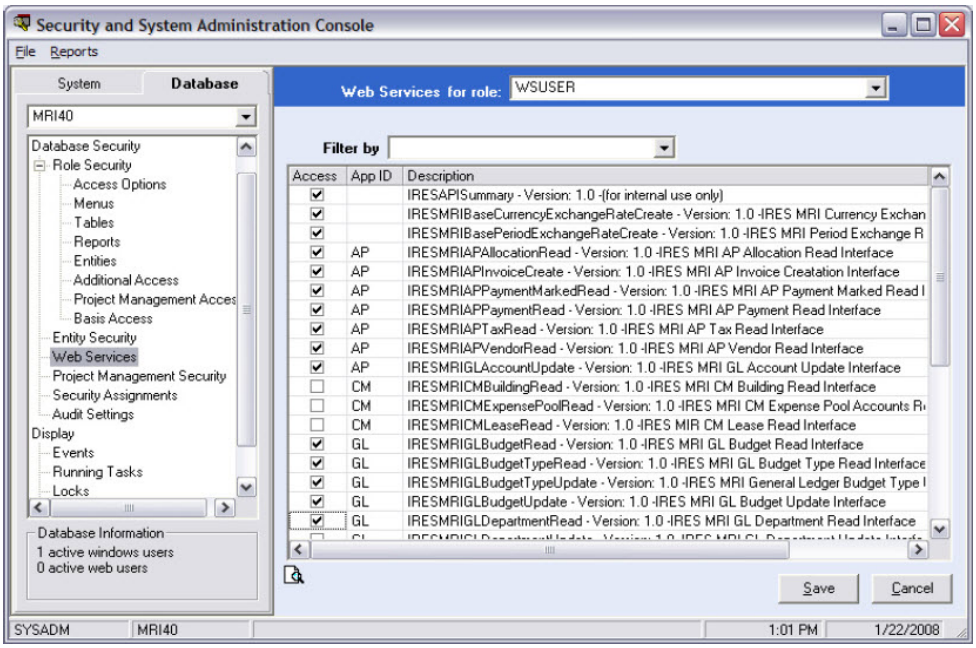

Figure 2-1. Possible API Assignments for Web Services User Role

Web Services users must also be assigned to the Web Services User role. In Figure 2-2, the Web Services User role is named WSUSER.

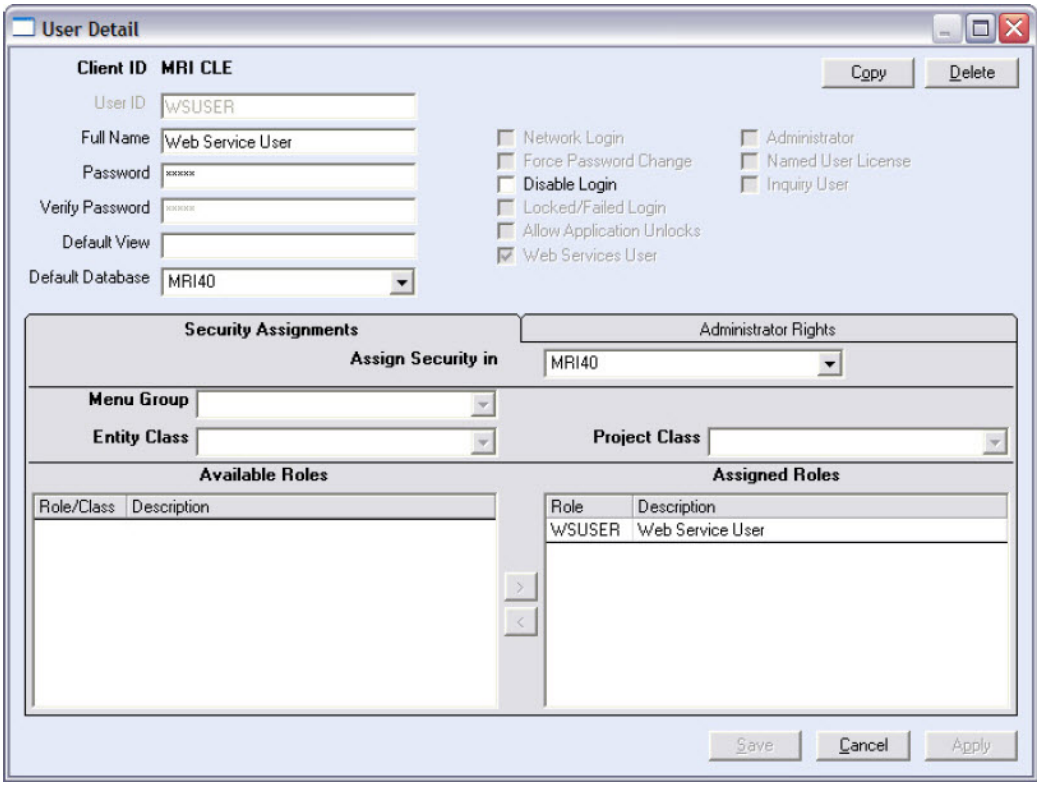

Figure 2-2. Web Services User Role

MRI Web Services Description

MRI Web Services (MRIAPIServices) can be used to share data between an MRI database and third-party applications. Depending on your MRI for the Web installation, the Web Services URL used to invoke exposed functions will be one of the following paths, where **[MRIWebDomain]** is the domain name of your MRI for the Web installation:

- If you installed MRI for the Web to the default website, the Web Services URL is https://[MRIWebDomain]/MriWeb/MRIAPIServices/APIService.asmx

  Note

  This URL will be used for examples throughout this document.

- If you installed MRI for the Web to a newly created website, the Web Services URL is https://[MRIWebDomain]/MRIAPIServices/APIService.asmx

No support exists for message encryption in Web Services. All encryption is done through the use of secured connections and SSL.

In MRI version 4.1 and later, field-level encryption is used in MRI databases. The data in encrypted fields is in binary format created using SQL encryption. Encrypted fields are not supported by MRI APIs.

Getting Started as a SaaS Client

If you are a SaaS client, you will need to contact your account representative to request a third-party integration. Your account representative will arrange any contract or billing changes necessary for the integration, and then an API Web Services user ID will be created for you.

Calling MRI APIs 3

MRI APIs can be invoked either as SOAP services or as plain XML over HTTP. Both mechanisms execute the same logic. It is up to the caller to decide which mechanism to use based on convenience, skill level, or available tooling.

This chapter assumes the reader has working knowledge of HTTP, XML, and SOAP. If you do not currently have a Web Services user set up, refer to "MRI Environment Prerequisites" on page 6.

Calling APIs as a Simple HTTP Request

The URL used to invoke an API over a simple HTTP request is shown below where **[MRIWebDomain]** is the domain name of your MRI for the Web installation, and **[APIName]** is the name of the API you want to invoke:

https://[MRIWebDomain]/MriWeb/MRIAPIServices/api/[APIName]

MRI uses basic HTTP authentication as defined by RFC 1945, section 11.1. Your user name and password will be provided to you by your MRI system administrator. For a description of the algorithm used to provide these credentials, refer to section 11.1 of the text file available at the following URL:

http://www.rfc-base.org/rfc-1945.html

As a general rule, your user name is structured as shown below, where **[ClientID]** is the ID of the MRI client you are calling and **[WebServicesUser]** is the user name of the Web Services user set up in "MRI Environment Prerequisites" on page 6:

[ClientID]/[Database]/[WebServicesUser]

For example, if your user name is P123456/DATABASE1/USER1 and your password is Password1!, you would send MRI the following request:

```
POST https://[MRIWebDomain]/MRIAPIServices/api/InterestingAPI
HTTP/1.1
Host: [MRIWebDomain]
Content-Type: application/xml
Authorization: Basic
UDEyMzQ1Ni9EQVRBQkFTRTEvVVNFUjE6UGFzc3dvcmQxIQ==
Content-Length: [length of body]

<?xml version="1.0" encoding="utf-8"?>
<InterestingAPI>
  <node>value</node>
  <node>value</node>
  <node>value</node>
</InterestingAPI>
```

MRI will respond on the same connection. For example, a successful response to the InterestingAPI example will look like the following:

```
HTTP/1.1 200 OK
Cache-Control: no-cache, no-store
Pragma: no-cache
Content-Length: [length of body]
Content-Type: application/xml
Expires: -1
Date: Sat, 24 May 2014 16:34:36 GMT

<InterestingAPIResponse>
  <node>value</node>
  <node>value</node>
</InterestingAPIResponse>
```

The body of the response is the direct output of the API.

In the case of an error, the response will contain the error message and number, if applicable. An appropriate HTTP status code will be provided based on the error to allow the calling code to react in a more predictable manner.

```
HTTP/1.1 401 Unauthorized
Cache-Control: private
Content-Length: [length of body]
Content-Type: text/html; charset=utf-8
WWW-Authenticate: Basic realm=""
Date: Sat, 24 May 2014 16:31:43 GMT

<InterestingAPIErrorMessage>
  <error>
    <message>
      User not authorized to access the requested API.
    </message>
    <errorNumber>
     110
    </errorNumber>
  </error>
</InterestingAPIErrorMessage>
```

In cases where some records were processed and some failed, meaning both a normal response and an error response are available, the response will contain both. For errors for specific APIs, refer to “Building-Block APIs” on page 17.

Calling APIs as SOAP Services

Assuming MRI for the Web is installed at http://example.mrisoftware.com/mriweb, the API endpoint would likely be at the following address:

http://example.com/MRIAPIServices/APIService.asmx

Note

Contact your MRI system administrator for the specific URL where your APIs are installed.

Your API endpoint is the same for all APIs. You will need to specify which API to invoke inside the body of the request. For example, to invoke an API named InterestingAPI, send the following HTTP request (critical headers are in bold):

```
POST http://example.com/MRIAPIServices/APIService.asmx HTTP/1.1
User-Agent: ...
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://realestate.intuit.com/IRESAPI/ExecuteAPICommand"
Host: example.com
Content-Length: ...
Expect: 100-continue

<?xml version="1.0" encoding="utf-8" ?>
<soap:envelope
 xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/
 xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
  <ExecuteAPICommand xmlns="http://realestate.intuit.com
   /IRESAPI/">
    <apiInfo>&lt;IRESAPI&gt;
     &lt;GUID&gt;MRI_WEBSERVICES_USERNAME&lt;/GUID&gt;
     &lt;GPWD&gt;MRI_WEBSERVICES_PASSWORD&lt;/GPWD&gt;
     &lt;CID&gt;MRI_CLIENT_ID&lt;/CID&gt;
     &lt;DBName&gt;APPLICATION_DATABASE_NAME&lt;/DBName&gt;
     &lt;APIName&gt;InterestingAPI&lt;/APIName&gt;
     &lt;/IRESAPI&gt;
    </apiInfo>
    <commandInfo>&lt;?xml encoding="utf-8" ?&gt;
     &lt;InterestingAPI&gt;
     &lt;node&gt;value&lt;/node&gt;
     &lt;node&gt;value&lt;/node&gt;
     &lt;node&gt;value&lt;/node&gt;
     &lt;/InterestingAPI&gt;
    </commandInfo>
    <returnedInfo />
    <errorInfo />
  </ExecuteAPICommand>
</soap:Body>
</soap:envelope>
```

apiInfo

The **apiInfo** node contains the following XML-encoded elements that describe the request and authentication:

Element	Description
<IRESAPI>	
<CID>string</CID>	Client ID
<DBName>string</DBName>	Database Name
<GUID>string</GUID>	Web Services User ID
<GPWD>string</GPWD>	Web Services User Password
<APIName>string</APIName>	API Name
<REQZIP>string</REQZIP>	Values: true/false; only valid for ExecuteAPICommandByte
<RESZIP>string</RESZIP>	Values: true/false; only valid for ExecuteAPICommandByte
</IRESAPI>	

The **REQZIP** node lets you specify that the request that you are sending is compressed. Conversely, the **RESZIP** node lets you request that the generated response from MRI be compressed. All compression is done according to the GZip standard.

commandInfo

The **commandInfo** node contains XML-encoded fragments of XML describing the useful payload of the request. Refer to “Building-Block APIs” on page 17 for the exact shape of this node for the API you want to call.

```
<InterestingAPI>
  <node>value</node>
  <node>value</node>
  <node>value</node>
</InterestingAPI>
```

MRI responds on the same connection. For example, the response to the InterestingAPI example would look like the following:

```
HTTP/1.1 200 OK
Cache-Control: private, max-age=0
Content-Length: 1560
Content-Type: text/xml; charset=utf-8
Date: Mon, 19 Dec 2011 20:17:10 GMT
```

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:envelope
 xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/
 xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ExecuteAPICommandResponse>
      <ExecuteAPICommandResult>
        <EXECUTED>FALSE</EXECUTED>
      </ExecuteAPICommandResult>
      <returnedInfo>
       &lt;InterestingAPIResponse&gt;
       &lt;node&gt;value&lt;/node&gt;
       &lt;node&gt;value&lt;/node&gt;
       &lt;node&gt;value&lt;/node&gt;
       &lt;/InterestingAPIResponse&gt;
      </returnedInfo>
      <errorInfo />
    </ExecuteAPICommandResponse>
  </soap:Body>
</soap:envelope>
```

returnedInfo

The **returnedInfo** node contains an XML-encoded response. Refer to “Building-Block APIs” on page 17 for the exact shape of the response.

```
<InterestingAPIResponse>
  <node>value</node>
  <node>value</node>
  <node>value</node>
</InterestingAPIResponse>
```

errorInfo

The **errorInfo** node is a string returned by the service function containing any errors or warnings that occurred while processing your request. For more information about the XML elements for a specific API, refer to “Building-Block APIs” on page 17.

Note

Some error response records can have both successful responses and error responses. You should review the errors you receive to determine whether it is a critical error or a warning. For more information about the severity of the errors, refer to the documentation for each API.

```
<InterestingAPIErrorMessage>
  <error>
    <message>
      <InterestingAPIError>
        <error>
          {API specific qualifier elements}
```

```
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </InterestingAPIError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
</InterestingAPIErrorMessage>
```

Calling APIs Asynchronously

If the APIs you are trying to call are taking a long time or timing out, consider using the asynchronous execution mode. Some APIs are only executable asynchronously. If you are attempting to execute an asynchronous API as synchronous, you will receive an error.

To execute an API asynchronously, append the **$async** query string parameter to the end of the URL. This query string parameter can be set to **prefer** or **demand**.

- **prefer**—This value implies that the caller would like to execute the API asynchronously, but will accept the response immediately if the API prefers not to be called asynchronously.
- **demand**—This value implies that the caller would like to execute the API asynchronously, but would rather the request fail if the API prefers not to be called asynchronously.

Other than the query string parameter, the calling procedure is the same as synchronous invocation, which is described in the earlier sections of this chapter.

Example

```
POST https://[MRIWebDomain]/MRIAPIServices/api/
             IRESMRICMLeaseRead?$async=demand HTTP/1.1
Host: [MRIWebDomain]
Content-Type: application/xml
Authorization: Basic UDEyMzQ1Ni9EQVRBQkFTRTEvVVNFUjE6UGFzc3dvcmQxIQ==
Content-Length: [length of body]

<?xml version="1.0" encoding="utf-8"?>
<IRESMRICMLeaseRead>
  <BLDGID>B1</BLDGID>
</IRESMRICMLeaseRead>
```

Note

The URL in this example has a line break for readability.

The server will respond with the body of the response if the API was executed synchronously, or with the following message specifying the ID of the scheduled request if the API was executed asynchronously:

```
<Scheduled>
  <Request ID='39e2dcd0-8d0a-46ae-87f7-0f732f015eed'/>
</Scheduled>
```

To retrieve the body of the response, call the AsyncApiStatusCheck API and send the body of the **Scheduled** response you received above. You can include multiple **Request** nodes within the **Scheduled** node to get the statuses of multiple asynchronous requests.

```
<Scheduled>
  <Request ID='39e2dcd0-8d0a-46ae-87f7-0f732f015eed'/>
</Scheduled>
```

The response of the AsyncApiStatusCheck API looks similar to the following response:

```
<?xml encoding="uft-8"?>
<AsyncApiStatusCheckResponse AsOf="2014-03-16T...">
  <Status RequestID="39e2dcd0-8d0a-46ae-87f7-0f732f015eed"
          Part="1"
          TotalParts="1"
          DateSubmitted="2014-03-16T..."
          DateStarted="2014-03-16T..."
          DateCompleted="2014-03-16T...">
    <Response>
      <![CDATA[<IRESMRICMLeaseReadResponse>
        <interfaceVersion>1.0</interfaceVersion>
        <rowsReturned>0</rowsReturned>
      </IRESMRICMLeaseReadResponse>]]>
    </Response>
    <Errors><![CDATA[any errors]]></Errors>
  </Status>
</AsyncApiStatusCheckResponse>
```

If the request was partitioned before scheduling, the response will contain multiple **Status** nodes with the same **RequestID** for each part.

Asynchronous Processing Callback

To be notified when your request finishes processing, you can specify the **$callbackurl** query string parameter. The system will send the status of the request to that URL as an HTTP POST request after the system has finished processing. The following is an example of this request, with new lines introduced for readability:

```
http://example.com/apiservice.asmx
     ?$async=demand
     &$callbackurl=http%3A%2F%2Fexample.com%2Freceivemriresponse
```

The callback will be an HTTP POST request to the callback URL, and the body will look similar to the following:

```
<AsyncCallback
  RequestID="39e2dcd0-8d0a-46ae-87f70f732f015eed"
  Part="1"
  TotalParts="1">
```

```
  <Response>
    <![CDATA[<IRESMRICMLeaseReadResponse>
      <interfaceVersion>1.0</interfaceVersion>
      <rowsReturned>0</rowsReturned>
    </IRESMRICMLeaseReadResponse>]]>
  </Response>
  <Errors><![CDATA["any errors"]]></Errors>
</AsyncCallback>
```

Building-Block APIs 4

This chapter provides detailed descriptions of all available integration points to be invoked by service providers. MRI APIs are passive functions. Once they are set up, they wait for a third party to make an API request to read or import data.

Base APIs

Currency Exchange Rate Create Interface (IRESMRIBaseCurrencyExchangeCreate)

This method provides the ability to create currency exchange rate entries in an MRI database. This method is available for all installations and provides for the creation of new standard currency exchange rate entries. No support is provided through this method for updating existing rates. Additionally, no support is provided for creating spot exchange rates. The request will **not** be treated as a batch. As a result, all valid rows will be processed, and the database will be updated with errors returned for unsuccessful items. This method provides support for additional custom fields, foreign key validation, and defaulting based on the definition in the MRIField table. The capitalized field name will be used as the tag name for all nodes unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

```
<IRESMRIBaseCurrencyExchangeCreate>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <CURREXCH>
    <EXCHGREF>string</EXCHGREF>                 Blocked
    <CURRCODE1>string</CURRCODE1>               Required
    <CURRCODE2>string</CURRCODE2>               Required
    <EXCHDATE>datetime</EXCHDATE>               Required
    <EXCHRATE>numeric</EXCHRATE>                Required
    <RATESOURCE>string</RATESOURCE>             Optional
    <LASTDATE>datetime</LASTDATE>               Blocked
    <USERID>string</USERID>                     Blocked
    <EXCHTYPE>string</EXCHTYPE>                 Blocked
    <EXCHCOUNTER>numeric</EXCHCOUNTER>          Blocked
  </CURREXCH>
  …
</IRESMRIBaseCurrencyExchangeCreate>
```

Note

- EXCHGREF is system generated.
- LASTDATE and USERID are updated using the current time and authenticated user ID.
- EXCHTYPE is defaulted to the value of 0.

returnedInfo (API response)

```
<IRESMRIBaseCurrencyExchangeCreateResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsInserted>numeric</rowsInserted>
  <rowsValidated>numeric</rowsValidated>
  <rowsInError>numeric</rowsInError>
</IRESMRIBaseCurrencyExchangeCreateResponse>
```

errorInfo (error response)

```
<IRESMRIBaseCurrencyExchangeCreateErrorMessage>
  <error>
    <message>
      <IRESMRIBaseCurrencyExchangeCreateError>
        <error>
          <CURRCODE1>string</CURRCODE1>          Where available
          <CURRCODE2>string</CURRCODE2>          Where available
          <EXCHDATE>datetime</EXCHDATE>          Where available
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIBaseCurrencyExchangeCreateError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIBaseCurrencyExchangeCreateErrorMessage>
```

Period Exchange Rate Create Interface (IRESMRIBasePeriodExchangeCreate)

This method provides the ability to create period exchange rate entries in an MRI database. This method is available for all installations and provides for the creation of new standard period exchange rate entries. No support is provided through this method for updating existing rates. The request will **not** be treated as a batch. As a result, all valid rows will be processed, and the database will be updated with errors returned for unsuccessful items. This method will provide support for additional custom fields, foreign key validation, and defaulting based on the definition in the MRIField table. The capitalized field name will be used as the tag name for all nodes unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

```
<IRESMRIBasePeriodExchangeCreate>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <PRDEXCH>
    <PEXCHGREF>string</PEXCHGREF>               Blocked
    <EXCHTYPE>string</EXCHTYPE>                 Required
    <CURRCODE1>string</CURRCODE1>               Required
    <CURRCODE2>string</CURRCODE2>               Required
    <EXCHPERIOD>string</EXCHPERIOD>             Required
    <EXCHRATE>numeric</EXCHRATE>                Required
    <RATESOURCE>string</RATESOURCE>             Optional
    <LASTDATE>datetime</LASTDATE>               Blocked
    <USERID>string</USERID>                     Blocked
  </PRDEXCH>
  …
</IRESMRIBasePeriodExchangeCreate>
```

Note

- PEXCHGREF is system generated.
- LASTDATE and USERID are updated using the current time and authenticated user ID.

returnedInfo (API response)

```
<IRESMRIBasePeriodExchangeCreateResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsInserted>numeric</rowsInserted>
  <rowsValidated>numeric</rowsValidated>
  <rowsInError>numeric</rowsInError>
</IRESMRIBasePeriodExchangeCreateResponse>
```

errorInfo (error response)

```
<IRESMRIBasePeriodExchangeCreateErrorMessage>
  <error>
    <message>
      <IRESMRIBasePeriodExchangeCreateError>
        <error>
          <EXCHYPE>string</EXCHTYPE>            Where available
          <CURRCODE1>string</CURRCODE1>         Where available
          <CURRCODE2>string</CURRCODE2>         Where available
          <EXCHPERIOD>string</EXCHPERIOD>       Where available
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIBasePeriodExchangeCreateError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIBasePeriodExchangeCreateErrorMessage>
```

Accounts Payable APIs

Allocation Read Interface (IRESMRIAPAllocationRead)

This method provides the ability to read accounts payable allocation information from the AP Allocations table (ALDE) and the AP Allocation Detail table (ALLO) in an MRI database. This method is available for all installations that include a license for the Accounts Payable application. All fields from the ALDE and ALLO tables, plus additional custom fields in either table, will be returned in the response. The capitalized field name will be used as the tag name for all nodes unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRIAPAllocationRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRIAPAllocationRead>
```

Overlay 2—A single row with subdetail for a specified allocation code.

```
<IRESMRIAPAllocationRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <ALLOCODE>string</ALLOCODE>                   Required
</IRESMRIAPAllocationRead>
```

Overlay 3—All rows updated since a specified last update date.

```
<IRESMRIAPAllocationRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LASTDATE>datetime</LASTDATE>                 Required
</IRESMRIAPAllocationRead>
```

returnedInfo (API response)

```
<IRESMRIAPAllocationReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>9999</rowsReturned>
  <ALDE>
    <ALLOCODE>string</ALLOCODE>
    <ALLODESC>string</ALLODESC>
    <LASTDATE>datetime</LASTDATE>
    <USERID>string</USERID>
    …{additional custom fields}
    <ALLO>
      <ALLOCODE>string</ALLOCODE>
```

```
      <ITEM>numeric</ITEM>
      <ENTITYID>string</ENTITYID>
      <DEPARTMENT>string</DEPARTMENT>
      <ACCTNUM>string</ACCTNUM>
      <PERCENT>numeric</PERCENT>
      <ATAXGRPID>string</ATAXGRPID>
      <JOBCODE>string</JOBCODE>
      <DISTRBTYPEID>string</DISTRBTYPEID>
      …{additional custom fields}
    </ALLO>
    …
  </ALDE>
  …
</IRESMRIAPAllocationReadResponse>
```

errorInfo (error response)

```
<IRESMRIAPAllocationReadErrorMessage>
  <error>
    <message>
      <IRESMRIAPAllocationReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIAPAllocationReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIAPAllocationReadErrorMessage>
```

Tax Read Interface (IRESMRIAPTaxRead)

This method provides the ability to read accounts payable tax information from the !{Sales Tax} Groups table (ATAXGRP), the !{Sales Tax} Group Maps table (ATAXMAP), and the !{Sales Tax} IDs table (ATAX) in an MRI database. This method is available for all installations that include a license for the Accounts Payable application. All fields from the ATAXGRP, ATAXMAP, and ATAX tables, plus additional custom fields in any of the three tables, will be returned in the response. The capitalized field name will be used as the tag name for all nodes unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRIAPTaxRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRIAPTaxRead>
```

Overlay 2—A single row with sub details for a specified tax group code.

```
<IRESMRIAPTaxRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <ATAXGRPID>string</ATAXGRPID>                 Required
</IRESMRIAPTaxRead>
```

returnedInfo (API response)

```
<IRESMRIAPTaxReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>999</rowsReturned>
  <ATAXGRP>
    <ATAXGRPID>string</ATAXGRPID>
    <DESCRPTN>string</DESCRPTN>
    <componentTax>
      <ATAXID>string</ATAXID>
      <TAXONTAX>string</TAXONTAX>
      <DESCRPTN>string</DESCRPTN>
      <ATAXPERC>numeric</ATAXPERC>
    </componentTax>
    …
  </ATAXGRP>
  …
</IRESMRIAPTaxReadResponse>
```

errorInfo (error response)

```
<IRESMRIAPTaxReadErrorMessage>
  <error>
    <message>
      <IRESMRIAPTaxReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIAPTaxReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIAPTaxReadErrorMessage>
```

Vendor Read Interface (IRESMRIAPVendorRead)

This method provides the ability to read accounts payable vendor and vendor alternate address information from the !{Vendors} table (VEND) and the Alternate !{Vendor} Payment Addresses table (TB_AP_ALTADDR) in an MRI database. This method is available for all installations that include a license for the Accounts Payable application. Only those fields specified, plus additional custom fields in the VEND table, will be provided in the response. The capitalized field name will be used as the tag name for all nodes unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRIAPVendorRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRIAPVendorRead>
```

Caution!
Due to the large number of vendors in some databases, clients may experience time-outs when using this overlay. When those situations occur, either the time-out needs to be adjusted or the process changed to use Overlay 4 to run for each letter and number.

Overlay 2—A single row with sub detail for a specified vendor ID.

```
<IRESMRIAPVendorRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <VENDID>string</VENDID>                       Required
</IRESMRIAPVendorRead>
```

Overlay 3—All rows updated since a specified last update date.

```
<IRESMRIAPVendorRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LASTDATE>datetime</LASTDATE>                 Required
</IRESMRIAPVendorRead>
```

Overlay 4—All rows with a vendor ID beginning with the specified value.

```
<IRESMRIAPVendorRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <beginningWith>string</beginningWith>         Required
</IRESMRIAPVendorRead>
```

Caution!
Due to the large number of vendors in some databases, clients may experience time-outs when using this overlay with a historic date. When those situations occur, either the time-out needs to be adjusted or the process changed to use Overlay 4 to run for each letter and number.

returnedInfo (API response)

```
<IRESMRIAPVendorReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>999</rowsReturned>
  <VEND>
    <VENDID>string</VENDID>
    <VENDNME1>string</VENDNME1>
    <VENDNME2>string</VENDNME2>
    <VENDADDR>string</VENDADDR>
    <CITY>string</CITY>
    <STATE>string</STATE>
    <ZIP>string</ZIP>
    <COUNTY>string</COUNTY>
    <M_1099REQ>string</M_1099REQ>
    <USEDISC>string</USEDISC>
    <LASTDATE>datetime</LASTDATE>
    <USERID>string</USERID>
    <FOREIGNVND>string</FOREIGNVND>
    <PHONE>string</PHONE>
    <STATUS>string</STATUS>
    <PAYTYPE>string</PAYTYPE>
    <REF>string</REF>
    <DELFLAG>string</DELFLAG>
    <ACCTNUM>string</ACCTNUM>
    <CONTACT>string</CONTACT>
    <TERMS>string</TERMS>
    <WIRETOCM>string</WIRETOCM>
    <HOMEONLY>string</HOMEONLY>
    <APPLIMIT>numeric</APPLIMIT>
    <VENDTYPE>string</VENDTYPE>
    <INSEXP>Date</INSEXP>
    <CURRCODE>string</CURRCODE>
    <SITEID>string</SITEID>
    <SUBCONT>string</SUBCONT>
    <CERTREQD>string</CERTREQD>
    <SUBWITH>string</SUBWITH>
    <PRIORITYID>string</PRIORITYID>
    <ATTNFEE>string</ATTNFEE>
    <EXPCERTOK>string</EXPCERTOK>
    <NOPOOK>string</NOPOOK>
    <FEDID>string</FEDID>
    …{Additional custom field nodes}
    <TB_AP_ALTADDR>
      <ALTADDRID>string</ALTADDRID>
      <VENDNME2>string</VENDNME2>
      <ADDR>string</ADDR>
      <CITY>string</CITY>
      <STATE>string</STATE>
      <ZIP>string</ZIP>
    </TB_AP_ALTADDR>
    …
  </VEND>
  …
</IRESMRIAPVendorReadResponse>
```

errorInfo (error response)

```
<IRESMRIAPVendorReadErrorMessage>
  <error>
    <message>
      <IRESMRIAPVendorReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIAPVendorReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIAPVendorReadErrorMessage>
```

Vendor Creation Interface (IRESMRIAPVendorCreate)

This method provides the ability to create new accounts payable vendors in the VEND table of an MRI database, but does not support the ability to update existing vendors. This method is available for all installations that include a license for the Accounts Payable application.

By default, the request will **not** be treated as a batch. As a result, all valid rows will be processed, and the database will be updated with errors returned for unsuccessful items. If the value of the **treatAsBatch** node is set to **Y**, the request will be treated as a batch. As a result, any error will cause a rejection of the whole request.

This method will provide support for additional custom fields, foreign key validation, and defaulting based on the definition in the MRIField table. If no input is received for optional tags and that field has a default value in MRIField, the default will be entered. Otherwise, user input will override the default. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

```
<IRESMRIAPVendorCreate>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <treatAsBatch>Y/N</treatAsBatch>              Optional
  <VEND>
    <VENDID>string</VENDID>                     Required
    <VENDNME1>string</VENDNME1>                 Required
    <VENDNME2></VENDNME2>                       Optional
    <VENDADDR></VENDADDR>                       Optional
    <VENDADDR2></VENDADDR2>                     Optional
    <CITY></CITY>                               Optional
    <STATE></STATE>                             Optional
```

```
    <ZIP></ZIP>                                          Optional
    <COUNTRY></COUNTRY>                                  Optional
    <FEDIDTYP></FEDIDTYP>                                Optional
    <FEDID></FEDID>                                      Optional
    <M_1099REQ></M_1099REQ>                              Optional
    <USEDISC></USEDISC>                                  Optional
    <ACCTNUM></ACCTNUM>                                  Optional
    <DELFLAG></DELFLAG>                                  Optional
    <DAYS1></DAYS1>                                      Optional
    <DISC1></DISC1>                                      Optional
    <DAYS2></DAYS2>                                      Optional
    <DISC2></DISC2>                                      Optional
    <DAYS3></DAYS3>                                      Optional
    <DISC3></DISC3>                                      Optional
    <LASTDATE></LASTDATE>                                Blocked
    <USERID></USERID>                                    Blocked
    <FOREIGNVND></FOREIGNVND>                            Optional
    <PHONE></PHONE>                                      Optional
    <STATUS></STATUS>                                    Optional
    <PAYTYPE></PAYTYPE>                                  Required
    <REF></REF>                                          Optional
    <VATREG></VATREG>                                    Optional
    <COUNTY></COUNTY>                                    Optional
    <ENTITYRESTR></ENTITYRESTR>                          Optional
    <CONTACT></CONTACT>                                  Optional
    <TERMS></TERMS>                                      Optional
    <WIRETOCM></WIRETOCM>                                Optional
    <HOMEONLY></HOMEONLY>                                Optional
    <APPLIMIT></APPLIMIT>                                Optional
    <VENDTYPE></VENDTYPE>                                Optional
    <INSEXP></INSEXP>                                    Optional
    <CURRCODE></CURRCODE>                                Optional
    <SITEID></SITEID>                                    Optional
    <SUBCONT></SUBCONT>                                  Optional
    <CERTREQD></CERTREQD>                                Optional
    <SUBWITH></SUBWITH>                                  Optional
    <PRIORITYID></PRIORITYID>                            Optional
    <ATTNFEE></ATTNFEE>                                  Optional
    <EXPCERTOK></EXPCERTOK>                              Optional
    <NOPOOK></NOPOOK>                                    Optional
    <CheckPrintingPriorityID></CheckPrintingPriorityID>  Optional
    <LASTEXPORTDATE></LASTEXPORTDATE>                    Optional
    <RMINTERNATIONAL></RMINTERNATIONAL>                  Optional
    …{additional custom fields tags}                     Optional
  </VEND>
  …
</IRESMRIAPVendorCreate>
```

Note

- FEDIDTYP, M_1099REQ, USEDISC, DELFLAG, FOREIGNVEND, STATUS, ENTITYRESTR, WIRETOCM, HOMEONLY, SUBCONT, CERTREQD, SUBWITH, PRIORITYID, ATTNFEE, EXPCERTOK, and NOPOOK are set based on their definition in the MRIField table if excluded. If any of these tags are included, the user input will override the MRIField default.

- LASTDATE and USERID are updated using the current time and authenticated user ID.

returnedInfo (API response)

```
<IRESMRIAPVendorCreateResponse>
   <interfaceVersion>1.0</interfaceVersion>
   <rowsInserted>numeric</rowsInserted>
   <rowsValidated>numeric</rowsValidated>
   <rowsInError>numeric</rowsInError>
</IRESMRIAPVendorCreateResponse>
```

errorInfo (error response)

```
<IRESMRIAPVendorCreateErrorMessage>
  <error>
    <message>
      <IRESMRIAPVendorCreateError>
        <error>
          <VENDID>string</VENDID>                Where Available
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIAPVendorCreateError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
</IRESMRIAPVendorCreateErrorMessage>
```

Vendor Update Interface (IRESMRIAPVendorUpdate)

This method provides the ability to update existing account payable vendor information in the VEND table of an MRI database. This method is available for all installations that include a license for the Accounts Payable application.

This method allows for the updating of existing accounts payable vendors and provides no support for creating new vendors. By default, the request will **not** be treated as a batch. As a result, all valid rows will be processed, and the database will be updated with errors returned for unsuccessful items. If the **treatAsBatch** tag is set to **Y**, the request will be treated as a batch. As a result, any error will cause a rejection of the whole request.

This method will provide support for additional custom fields and foreign key validation. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

```
<IRESMRIAPVendorUpdate>
  <interfaceVersion>1.0</interfaceVersion>          Optional
  <treatAsBatch>Y/N</treatAsBatch>                  Optional
  <VEND>
    <VENDID>string</VENDID>                         Required
    <VENDNME1>string</VENDNME1>                     Required
    <VENDNME2></VENDNME2>                           Optional
    <VENDADDR></VENDADDR>                           Optional
    <VENDADDR2></VENDADDR2>                         Optional
    <CITY></CITY>                                   Optional
    <STATE></STATE>                                 Optional
    <ZIP></ZIP>                                     Optional
    <COUNTRY></COUNTRY>                             Optional
    <FEDIDTYP></FEDIDTYP>                           Optional
    <FEDID></FEDID>                                 Optional
    <M_1099REQ></M_1099REQ>                         Optional
    <USEDISC></USEDISC>                             Optional
    <ACCTNUM></ACCTNUM>                             Optional
    <DELFLAG></DELFLAG>                             Optional
    <DAYS1></DAYS1>                                 Optional
    <DISC1></DISC1>                                 Optional
    <DAYS2></DAYS2>                                 Optional
    <DISC2></DISC2>                                 Optional
    <DAYS3></DAYS3>                                 Optional
    <DISC3></DISC3>                                 Optional
    <LASTDATE></LASTDATE>                           Blocked
    <USERID></USERID>                               Blocked
    <FOREIGNVND></FOREIGNVND>                       Optional
    <PHONE></PHONE>                                 Optional
    <STATUS></STATUS>                               Optional
    <PAYTYPE></PAYTYPE>                             Optional
    <REF></REF>                                     Optional
    <VATREG></VATREG>                               Optional
    <COUNTY></COUNTY>                               Optional
    <ENTITYRESTR></ENTITYRESTR>                     Optional
    <CONTACT></CONTACT>                             Optional
    <TERMS></TERMS>                                 Optional
    <WIRETOCM></WIRETOCM>                           Optional
    <HOMEONLY></HOMEONLY>                           Optional
    <APPLIMIT></APPLIMIT>                           Optional
    <VENDTYPE></VENDTYPE>                           Optional
    <INSEXP></INSEXP>                               Optional
    <CURRCODE></CURRCODE>                           Optional
    <SITEID></SITEID>                               Optional
    <SUBCONT></SUBCONT>                             Optional
    <CERTREQD></CERTREQD>                           Optional
    <SUBWITH></SUBWITH>                             Optional
    <PRIORITYID></PRIORITYID>                       Optional
    <ATTNFEE></ATTNFEE>                             Optional
    <EXPCERTOK></EXPCERTOK>                         Optional
    <NOPOOK></NOPOOK>                               Optional
    <CheckPrintingPriorityID></CheckPrintingPriorityID>  Optional
    <LASTEXPORTDATE></LASTEXPORTDATE>               Optional
    <RMINTERNATIONAL></RMINTERNATIONAL>             Optional
    …{additional custom fields tags}                Optional
```

```
  </VEND>
  …
</IRESMRIAPVendorUpdate>
```

Note

LASTDATE and USERID are updated using the current time and authenticated user ID.

returnedInfo (API response)

```
<IRESMRIAPVendorUpdateResponse>
   <interfaceVersion>1.0</interfaceVersion>
   <rowsInserted>numeric</rowsInserted>
   <rowsValidated>numeric</rowsValidated>
   <rowsInError>numeric</rowsInError>
</IRESMRIAPVendorUpdateResponse>
```

errorInfo (error response)

```
<IRESMRIAPVendorUpdateErrorMessage>
  <error>
    <message>
      <IRESMRIAPVendorUpdateError>
        <error>
          <VENDID>string</VENDID>                Where Available
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIAPVendorUpdateError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
</IRESMRIAPVendorUpdateErrorMessage>
```

Vendor Alternate Address Creation Interface (IRESMRIAPVendorAltAddressCreate)

This method provides the ability to create alternate addresses for vendors in the TB_AP_ALTADDR table of an MRI database. This method is available for all installations that include a license for the Accounts Payable application.

This method allows for the creation of new alternate addresses for existing vendors and provides no support for updating existing alternate addresses. By default, the request will **not** be treated as a batch. As a result, all valid rows will be processed, and the database will be updated with errors returned for unsuccessful items. If the **treatAsBatch** tag is set to **Y**, the request will be treated as a batch. As a result, any error will cause a rejection of the whole request.

This method will provide support for additional custom fields, foreign key validation, and defaulting based on the definition in the MRIField table. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

```
< IRESMRIAPVendorAltAddressUpdate>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <treatAsBatch>Y/N</treatAsBatch>              Optional
  <VENDALT>
    <VENDID>string</VENDID>                     Req'd for Insert
    <ALTADDRID>string</ALTADDRID>               Required
    <VENDNME2>string</VENDNME2>                 Optional
    <ADDR>string</ADDR>                         Optional
    <CITY>string</CITY>                         Optional
    <STATE>string</STATE>                       Optional
    <ZIP>string</ZIP>                           Optional
    {additional custom fields tags>             Optional
  </VENDALT>
</IRESMRIAPVendorAltAddressCreate>
```

Note

VENDID must exist in the VEND table in order to create a new entry in the TB_AP_ALTADDR table.

returnedInfo (API response)

```
<IRESMRIAPVendorAltAddressCreateResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsInserted>numeric</rowsInserted>
  <rowsValidated>numeric</rowsValidated>
  <rowsInError>numeric</rowsInError>
</IRESMRIAPVendorAltAddressCreateResponse>
```

errorInfo (error response)

```
<IRESMRIAPVendorAltAddressCreateErrorMessage>
  <error>
    <message>
      <IRESMRIAPVendorAltAddressCreateError>
        <error>
          <VENDID>string</VENDID>               Where Available
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIAPVendorAltAddressCreateError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
</IRESMRIAPVendorAltAddressCreateErrorMessage>
```

Vendor Alternate Address Update Interface (IRESMRIAPVendorAltAddressUpdate)

This method provides the ability to update alternate addresses for vendors in the TB_AP_ALTADDR table of an MRI database. This method is available for all installations that include a license for the Accounts Payable application.

This method allows for the updating of existing alternate addresses and provides no support for creating new alternate addresses. By default, the request will **not** be treated as a batch. As a result, all valid rows will be processed, and the database will be updated with errors returned for unsuccessful items. If the **treatAsBatch** tag is set to **Y**, the request will be treated as a batch. As a result, any error will cause a rejection of the whole request.

This method will provide support for additional custom fields, foreign key validation, and defaulting based on the definition in the MRIField table. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

```
<IRESMRIAPVendorAltAddressUpdate>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <treatAsBatch>Y/N</treatAsBatch>              Optional
  <VENDALT>
    <VENDID>string</VENDID>                     Required
    <ALTADDRID>string</ALTADDRID>               Required
    <VENDNME2>string</VENDNME2>                 Optional
    <ADDR>string</ADDR>                         Optional
    <CITY>string</CITY>                         Optional
    <STATE>string</STATE>                       Optional
    <ZIP>string</ZIP>                           Optional
    {additional custom fields tags>             Optional
  <VENDALT>
<IRESMRIAPVendorAltAddressUpdate>
```

returnedInfo (API response)

```
<IRESMRIAPVendorAltAddressUpdateResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsInserted>numeric</rowsInserted>
  <rowsValidated>numeric</rowsValidated>
  <rowsInError>numeric</rowsInError>
</IRESMRIAPVendorAltAddressUpdateResponse>
```

errorInfo (error response)

```
<IRESMRIAPVendorAltAddressUpdateErrorMessage>
  <error>
    <message>
      <IRESMRIAPVendorAltAddressUpdateError>
        <error>
          <VENDID>string</VENDID>                Where Available
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIAPVendorAltAddressUpdateError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
</IRESMRIAPVendorAltAddressUpdateErrorMessage>
```

Invoice Creation Interface (MRIAPInvoiceCreate)

This method provides the ability to create accounts payable invoices in the Invoice Header table (INVC) and the Invoice History table (HIST) of an MRI database. This method is available for all installations that include a license for the Accounts Payable application.

This method allows for the creation of new accounts payable invoices and provides no support for updating existing invoices. The request will be treated as a batch by default. As a result, any error will result in a rejection of the whole request. If the **process-as-batch** tag is set to **N**, the request will not be treated as a batch.

The **InterfaceId** field on the INVC table will be updated with a system-generated value, and that value will be returned as part of the response. If the request contains the optional **clientBatchMarker** tag, the INVC table will be updated with that value as well.

This method will provide support for additional custom fields, foreign key validation, and defaulting based on the definition in the MRIField table. All business logic and validation built into the MRIAPXML.DLL will be enforced, but no additional validation with will be added as part of this task. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

```
<MRIAPInvoiceCreate>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <process-as-batch>Y</process-as-batch>        Optional
  <SESSIONDESC>string</SESSIONDESC>             Optional
  <clientBatchMarker>string</clientBatchMarker> Optional
  <INVC>                                        Required
```

<VENDID>string</VENDID>	**Required**
<INVOICE>string</INVOICE>	**Required**
<EXPPED>string</EXPPED>	**Required**
<INVCDATE>datetime</INVCDATE>	Optional
<DUEDATE>datetime</DUEDATE>	Optional
<INVCAMT>numeric</INVCAMT>	Optional
<PAIDAMT>numeric</PAIDAMT>	**Blocked**
<SEPCHECK>string</SEPCHECK>	Optional
<PONUM>string</PONUM>	Optional
<LASTDATE>datetime</LASTDATE>	**Blocked**
<SESSION>string</SESSION>	**Blocked**
<USERID>string</USERID>	**Blocked**
<PARTIALPRG>string</PARTIALPRG>	Optional
<ATAXAMT>numeric</ATAXAMT>	Optional
<INVCTOT>numeric</INVCTOT>	Optional
<STATUS>string</STATUS>	Optional
<AUTHDATE>datetime</AUTHDATE>	Optional
<AUTHLOC>string</AUTHLOC>	Optional
<AUTHID>string</AUTHID>	Optional
<ECURCODE>string</ECURCODE>	Optional
<PLANID>string</PLANID>	Optional
<SITEID>string</SITEID>	Optional
<PROTYPE>string</PROTYPE>	Optional
<PBEGDATE>datetime</PBEGDATE>	Optional
<PENDDATE>datetime</PENDDATE>	Optional
<INCLATAXAMT>numeric</INCLATAXAMT>	**Blocked**
<JOBCOST>string</JOBCOST>	Optional
<CROSSSITE>string</CROSSSITE>	**Blocked**
<ALLOCODE>string</ALLOCODE>	Optional
<VENDID2ND>string</VENDID2ND>	Optional
<DESCRPTN>string</DESCRPTN>	Optional
<OUTSIDEBUD>string</OUTSIDEBUD>	Optional
<TAXPTDATECALC>datetime</TAXPDATECALC>	Optional
<EXPCTRLFLAG>string</EXPCTRLFLAG>	Optional
<APINVCTYPEID>string</APINVCTYPEID>	Optional
<OWNTBL>string</OWNTBL>	Optional
<OWNYEAR>string</OWNYEAR>	Optional
<OWNNUM>numeric</OWNNUM>	**Blocked**
<CTRYTBL>string</CTRYTBL>	Optional
<CTRYYEAR>string</CTRYYEAR>	Optional
<CTRYNUM>numeric</CTRYNUM>	**Blocked**
<APIMPORTNUM>string</APIMPORTNUM>	**Blocked**
<PERIODDATE>datetime</PERIODDATE>	Optional
<SCANNEDCOPYURL>string</SCANNEDCOPYURL>	Optional
…{additional custom field tags}	Optional
<HIST>	**Required**
<ITEM>numeric</ITEM>	Optional
<REF>string</REF>	Optional
<ENTITYID>string</ENTITYID>	**Required**
<ACCTNUM>string</ACCTNUM>	**Required**
<ITEMAMT>numeric</ITEMAMT>	Optional
<DISCAMT>numeric</DISCAMT>	Optional
<EXPPOST>string</EXPPOST>	Optional
<STATUS>string</STATUS>	Optional
<BALFLAG>string</BALFLAG>	Optional
<CHECKNO>string</CHECKNO>	Optional
<CHECKDT>datetime</CHECKDT>	Optional

<CHECKPD>string</CHECKPD>	Optional
<DISCTKN>string</DISCTKN>	**Blocked**
<CKPOSTED>string</CKPOSTED>	Optional
<CASHTYPE>string</CASHTYPE>	Optional
<JOBCODE>string</JOBCODE>	Optional
<POLINENMBR>numeric</POLINENMBR>	Optional
<ATAXID>string</ATAXID>	Optional
<ATAXAMT>numeric</ATAXAMT>	**Blocked**
<PONUM>string</PONUM>	Optional
<EXPGLREF>string</EXPGLREF>	**Blocked**
<DEPARTMENT>string</DEPARTMENT>	Optional
<TAXITEM>string</TAXITEM>	Optional
<BANKID>string</BANKID>	**Blocked**
<CKACCRLREF>string</CKACCRLREF>	Optional
<CKCASHGLREF>string</CKCASHGLREF>	Optional
<LASTDATE>datetime</LASTDATE>	**Blocked**
<USERID>string</USERID>	**Blocked**
<ATAXGRPID>string</ATAXGRPID>	Optional
<TAXMOD>string</TAXMOD>	Optional
<CURRCODE>string</CURRCODE>	Optional
<EXCHGREF>string</EXCHGREF>	Optional
<ECURCODE>string</ECURCODE>	Optional
<ECITEMAMT>numeric</ECITEMAMT>	Optional
<ECDISCAMT>numeric</ECDISCAMT>	Optional
<ECATAXAMT>numeric</ECATAXAMT>	Optional
<PCURCODE>string</PCURCODE>	Optional
<PCHECKAMT>numeric</PCHECKAMT>	Optional
<XITEMAMT>numeric</XITEMAMT>	Optional
<XDISCAMT>numeric</XDISCAMT>	Optional
<XEXCHGREF>string</XEXCHGREF>	Optional
<VOIDPD>string</VOIDPD>	Optional
<ORIGITEMAMT>numeric</ORIGITEMAMT>	Optional
<TRANSFERITEM>string</TRANSFERITEM>	Optional
<OWNERTAX>string</OWNERTAX>	Optional
<WITHATAXGRP>string</WITHATAXGRP>	Optional
<DISCMOD>string</DISCMOD>	**Blocked**
<SUBWITH>string</SUBWITH>	Optional
<WITHATAXID>string</WITHATAXID>	Optional
<INTENTEXPGLREF>string</INTENTEXPGLREF>	Optional
<INTENTCASHREF>string</INTENTCASHREF>	Optional
<INTENTACCRLREF>string</INTENTACCRLREF>	Optional
<POQTY>numeric</POQTY>	Optional
<JOBCOST>string</JOBCOST>	Optional
<JC_PHASECODE>string</JC_PHASECODE>	Optional
<JC_COSTLIST>string</JC_COSTLIST>	Optional
<JC_COSTCODE>string</JC_COSTCODE>	Optional
<CATEGORY>string</CATEGORY>	Optional
<CROSSSITE>string</CROSSSITE>	**Blocked**
<INVDRSTATUS>string</INVDRSTATUS>	Optional
<LOANID>string</LOANID>	Optional
<DRAWNO>string</DRAWNO>	Optional
<BATCHID>string</BATCHID>	Optional
<SRCBANKID>string</SRCBANKID>	Optional
<JCBILLTRAN>string</JCBILLTRAN>	Optional
<AUTAXPAID>atring</AUTAXPAID>	Optional
<RPTRUNID>string</RPTRUNID>	Optional
<DISTRBTYPEID>string</DISTRBTYPEID>	Optional

```
      <TAXPTDATECALC>datetime</TAXPDDATECALC>   Optional
      <TAXPTDATEACRLRPT>datetime</TAXPTDATEACRLRPT>   Optional
      <TAXPTDATECASHRPT>datetime</TAXPTDATECASHRPT>   Optional
      <RECEIPTCOMMITTED>string</RECEIPTCOMMITTED>     Optional
      <APINVCTYPEID>string</APINVCTYPEID>       Optional
      <RECEIPTNUMBER>string</RECEIPTNUMBER>     Optional
      <RECEIPTJOURNALREFERENCEACCRUAL />        Optional
      <RECEIPTJOURNALPERIODACCRUAL />           Optional
      <RECEIPTJOURNALREFERENCECASH />           Optional
      <RECEIPTJOURNALPERIODCASH />              Optional
      <PACKPROC />                              Blocked
      <PACKINIT />                              Blocked
      <PACKADJ />                               Blocked
      <PACKAGEID />                             Blocked
      <ADDLDESC>string</ADDLDESC>               Optional
      <CRELEDGID>string</CRELEDGID>             Blocked
      …{additional custom fields tags}          Optional
    </HIST>
  </INVC>
  …
</MRIAPInvoiceCreate>
```

Note

- SESSIONDESC will default to a standard built description if not provided.
- INVCDATE will default to system date if not provided. Be aware that if a date is provided, it must be formatted as YYYY-MM-DD.
- DUEDATE will be calculated from INVCDATE if not provided.
- INVCAMT will be calculated from the details if not provided.
- HIST: ITEM will be assigned if not provided.
- HIST: ACCTNUM will use the vendor default account number if not provided.
- HIST: STATUS will use the entity default status if not provided.
- HIST: CASHTYPE will use the entity default cash type if not provided.
- HIST: DEPARTMENT will be set to @ if not provided.
- Additional custom field tags will be mapped to custom fields in the INVC or HIST table by the field name. No additional validation will be performed beyond the foreign key validation with default values being set from the MRIFIELD definition where provided.

returnedInfo (API response)

```
<MRIAPInvoiceCreateResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsInserted>999</rowsInserted>
  <rowsValidated>numeric</rowsValidated>
  <rowsInError>numeric</rowsInError>
</MRIAPInvoiceCreateResponse>
```

errorInfo (error response)

```
<MRIAPInvoiceCreateErrorMessage>
  <error>
    <message>
      <MRIAPInvoiceCreateError>
        <error>
          <INVOICE>string</INVOICE>              Where available
          <EXPPED>string</EXPPED>                Where available
          <VENDID>string</VENDID>                Where available
          <ITEM>numeric</ITEM>                   Where available
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </MRIAPInvoiceCreateError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</MRIAPInvoiceCreateErrorMessage>
```

Payment Read Interface (MRIAPPaymentRead)

This method provides the ability to read AP Payment information from an MRI database. This method is available for all installations that include a license for the Accounts Payable application. Only those fields specified, plus additional custom fields in the !{Check} Summary table (SCHK), will be provided in the response. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All payments and voids for a specified vendor ID.

```
<MRIAPPaymentRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <VENDID>string</VENDID>                       Required
</MRIAPPaymentRead>
```

Overlay 2—All payments and voids since a specified last update date.

```
<MRIAPPaymentRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LASTDATE>datetime</LASTDATE>                 Required
</MRIAPPaymentRead>
```

Overlay 3—All payments and voids assigned to a specified check batch.

```
<MRIAPPaymentRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <BATCHID>string</BATCHID>                     Required
</MRIAPPaymentRead>
```

Overlay 4—All payments and voids made on a specified check date.

```
<MRIAPPaymentRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <CHECKDT>datetime</CHECKDT>                   Required
</MRIAPPaymentRead>
```

Overlay 5—All payments and voids for a specified bank ID since a specified last update date.

```
<MRIAPPaymentRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <BANKID>string</BANKID>                       Required
  <LASTDATE>datetime</LASTDATE>                 Required
</MRIAPPaymentRead>
```

Overlay 6—All payments and voids for a specified vendor ID since a specified last update date.

```
<MRIAPPaymentRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <VENDID>string</VENDID>                       Required
  <LASTDATE>datetime</LASTDATE>                 Required
</MRIAPPaymentRead>
```

returnedInfo (API response)

```
<MRIAPPaymentReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>999</rowsReturned>
  <SCHK>
    <BANKID>string</BANKID>
    <CHECKNO>string</CHECKNO>
    <VENDID>string</VENDID>
    <CHECKDT>datetime</CHECKDT>
    <CHECKPD>sring</CHECKPD>
    <CHECKNET>numeric</CHECKNET>
    <CKSTATUS>string</CKSTATUS>
    <LASTDATE>datetime</LASTDATE>
    <USERID>string</USERID>
    <DSTATCHG>datetime</DSTATCHG>
    <VENDNME1>string</VENDNME1>
    <ENTITYID>string</ENTITYID>
    <BATCHID>string</BATCHID>
    <VENDID2ND>string</VENDID2ND>
    <VENDNME2>string</VENDNME2>
    <VENDADDR>string</VENDADDR>
```

```
    <CITY>string</CITY>
    <STATE>string</STATE>
    <ZIP>string</ZIP>
    <DISTRBTYPEID>string</DISTRBTYPEID>
    …{additional custom field tags}
    <HIST>
      <INVOICE>string</INVOICE>
      <EXPPED>string</EXPPED>
      <ITEM>numeric</ITEM>
      <ITEMAMT>numeric</ITEMAMT>
      <DISCAMT>numeric</DISCAMT>
      <DISCTKN>string</DISCTKN>
      …{additional custom field tags}
    </HIST>
    …
  </SCHK>
  …
</MRIAPPaymentReadResponse>
```

errorInfo (error response)

```
<MRIAPPaymentReadErrorMessage>
  <error>
    <message>
      <MRIAPPaymentReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </MRIAPPaymentReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</MRIAPPaymentReadErrorMessage>
```

Payment Marked Read Interface (IRESMRIAPPaymentMarkedRead)

This method provides the ability to read unmarked accounts payable payment and voided payment information from the SCHK and HIST tables in an MRI database. This method is available for all installations that include a license for the Accounts Payable application. Unmarked records will be identified as any SCHK record where STATUS equals **O** and PAYINTERFACEID is null or where STATUS equals **V** and VOIDINTERFACEID is null. Marking will occur through the update of additional columns on the (SCHK) table named PAYINTERFACEID and VOIDINTERFACEID. The interface ID will be a system-generated value and will be included in the response. Marking will occur if the **markRows** tag is present. Otherwise, all unmarked rows will be returned, but marking will not be done and the interface ID will not be generated. If a **clientBatchMarker** tag is included in the request, the value of this tag will be used to update the SCHK table. Only those fields specified, plus additional custom fields in the SCHK table, will be

provided in the response. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

Note

When using the Payment Marked Read Interface, performance can be improved through the creation of the following additional indexes:

- **Invoice History Table (HIST)**—Add a non-unique index on PAYINTERFACEID and VOIDINTERFACEID.
- **Check Summary Table (SCHK)**—Add a non-unique index on INTERFACEID.

These indexes are not included with the standard installation due to the likelihood that they would slow the saving of journals and payments through the standard user interface.

commandInfo (request)

Overlay 1—All unmarked payments and voids.

```
<IRESMRIAPPaymentMarkedRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
  <markRows/>                                   Optional
  <clientBatchMarker>string</clientBatchMarker> Optional
</IRESMRIAPPaymentMarkedRead>
```

Overlay 2—All unmarked payments and voids for a specified bank ID.

```
<IRESMRIAPPaymentMarkedRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <BANKID>string</BANKID>                       Required
  <markRows/>                                   Optional
  <clientBatchMarker>string</clientBatchMarker> Optional
</IRESMRIAPPaymentMarkedRead>
```

returnedInfo (API response)

```
<IRESMRIAPPaymentMarkedReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>999</rowsReturned>
  <SCHK>
    <BANKID>string</BANKID>
    <CHECKNO>string</CHECKNO>
    <VENDID>string</VENDID>
    <CHECKDT>datetime</CHECKDT>
    <CHECKNET>numeric</CHECKNET>
    <CKSTATUS>string</CKSTATUS>
    <LASTDATE>datetime</LASTDATE>
    <USERID>string</USERID>
    <DSTATCHG>datetime</DSTATCHG>
    <VENDNME1>string<VENDNME1>
    <BATCHID>string</BATCHID>
```

```
    <VENDID2ND>string</VEND2ND>
    <VENDNME2>string</VENDNME2>
    <VENDADDR>string</VENDADDR>
    <CITY>string</CITY>
    <STATE>string</STATE>
    <ZIP>string</ZIP>
    <DISTRBTYPEID>string</DISTRBTYPEID>
    …{additional custom field tags}
    <HIST>
      <INVOICE>string</INVOICE>
      <EXPPED>string</EXPPED>
      <ITEM>numeric</ITEM>
      <ITEMAMT>numeric</ITEMAMT>
      <DISCAMT>numeric</DISCAMT>
      <DISCTKN>string</DISCTKN>
       …{additional custom field tags}
    </HIST>
    …
  </SCHK>
  …
</IRESMRIAPPaymentMarkedReadResponse>
```

errorInfo (error response)

```
<IRESMRIAPPaymentMarkedReadErrorMessage>
  <error>
    <message>
      <IRESMRIAPPaymentMarkedReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIAPPaymentMarkedReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIAPPaymentMarkedReadErrorMessage>
```

Commercial Management APIs

Building Read Interface (IRESMRICMBuildingRead)

This method provides the ability to read commercial building information from the !{Buildings} table (BLDG) in an MRI database. This method is available for all installations that include a license for the Commercial Management application. Only specified standard fields, plus additional custom fields, will be included in the response. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRICMBuildingRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRICMBuildingRead>
```

Overlay 2—A single row for the specified building ID.

```
<IRESMRICMBuildingRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <BLDGID>string</BLDGID>                       Required
</IRESMRICMBuildingRead>
```

Overlay 3—All rows updated since a specified last update date.

```
<IRESMRICMBuildingRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LASTDATE>datetime</LASTDATE>                 Required
</IRESMRICMBuildingRead>
```

returnedInfo (API response)

```
<IRESMRICMBuildingReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>999</rowsReturned>
  <BLDG>
    <BLDGID>string</BLDGID>
    <BLDGNAME>string</BLDGNAME>
    <ADDRESS1>string</ADDRESS1>
    <ADDRESS2>string</ADDRESS2>
    <CITY>string</CITY>
    <STATE>string</STATE>
    <ZIPCODE>string</ZIPCODE>
    <PHONENO>string</PHONENO>
    <LASTDATE>datetime</LASTDATE>
    <USERID>string</USERID>
    …{additional custom field tags}
  </BLDG>
  …
</IRESMRICMBuildingReadResponse>
```

errorInfo (error response)

```
<IRESMRICMBuildingReadErrorMessage>
  <error>
    <message>
      <IRESMRICMBuildingReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
```

```
        </error>
      <IRESMRICMBuildingReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRICMBuildingReadErrorMessage>
```

CM Charges (MRICMCharges)

This method provides the ability for external sources, such as Workspeed, to send CM charges to MRI. For a Workspeed installation, this method is available if the MRI Integration module is enabled and has the MRI Integration fields set up to connect to the MRI environment. Also, Workspeed users who are involved in the process must be set up to receive email notifications and have the ability to transfer to accounting.

After a batch of charges from the API is posted in MRI, MRI can launch a web event to notify the external source that the batch posted successfully. For this to occur, callback values must be specified in the request. Also, a port must be open in the MRI firewall for the outbound connection from MRI to Workspeed. The port to be opened is the port specified in the callback URL. The default ports are port 80 for nonsecure “http://” URLs and port 443 for secure “https://” URLs.

commandInfo (request)

The **commandInfo** node describes the contents of the request.

```
<MRICMCharges>
  <callbackURL>...</callbackURL>                Optional
  <callbackMimeType>...</callbackMimeType>      Optional
  <description>description of the batch
   of the charge to be created</description>    Optional
  <charges>                                     Required
    <![CDATA[...]]>
  </charges>
</MRICMCharges>
```

Note

- The **callbackURL** node contains the fully qualified URL to which the response will be posted once the batch is reviewed and posted. Any user name and password authentication must be included in the URL using the standard syntax, for example, http://username:password@api.workspeed.com/etc/etc/.
- The **callbackMimeType** node contains the MIME type to guide the system on the format of the response. In this case, the MIME type will be "application/vnd.workspeed.chargesTransferred" to get the response described below.

- The **charges** node contains comma-separated values describing the charges to be saved and includes the following data:
  - **CMBATCHID**—A string of variable size that is defined in MRI. The value must exist in MRI if specified. If the string is empty, a new batch ID will be created.
  - **ITEM**—A positive integer that uniquely identifies with the batch.
  - **SITEID**—A string of variable size that is defined in MRI and identifies the site associated with the transaction. The value must exist in MRI.
  - **BLDGID**—A string of variable size that is defined in MRI and identifies the building associated with the transaction. The value must exist in MRI.
  - **LEASID**—A string of variable size that is defined in MRI and identifies the lease associated with the transaction. The value must exist in MRI.
  - **TRANDATE**—The date of the transaction in ISO-8601 format, such as 2011-10-18 for October 18, 2011.
  - **INCCAT**—A string of variable size that is defined in MRI and identifies the income category associated with the transaction. The value must exist in MRI.
  - **SRCCODE**—A string of variable size that is defined in MRI and identifies the source code associated with the transaction. The value must exist in MRI. Typically, the value is **CH** for charge.
  - **DESCRPTN**—A string of variable size that is defined in MRI and describes the transaction.
  - **TRANAMT**—A positive, non-zero, decimal value that provides the transaction amount.
  - **RTAXAMT**—A positive decimal or zero value that provides the rent tax amount.
  - **RTAXGRPID**—A hardcoded string that identifies the rent tax group to which to record the tax amount. The hardcoded rent tax group ID must be an existing ID in MRI. If a charge amount in Workspeed includes a tax and the charge is transferred to MRI, MRI records the tax on a separate line item. The rent tax group ID is used to relate the separate tax line item to the charge line item.
  - **DEPARTMENT**—A string of variable size that is defined in MRI that identifies the department associated with the transaction. The value must exist in MRI.
  - **POSTORDER**—A numeric value that identifies the order in which the transaction is posted to the CM ledger.
  - **AUTOEXCEPTION**—A **Y** or **N** value for yes or no to indicate whether to automatically apply the transaction to existing prepayments when the batch is posted. This setting is only applicable if the **Allow Open Items to Match** option is selected in the CM management options. For

example, if the management option is selected and AUTOEXCEPTION is set to **Y**, then the charge will not be picked up for automatic application. If the management option is selected and AUTOEXCEPTION is set to **N**, then the charge will be picked up for automatic application.

returnedInfo (API response)

The **returnedInfo** node also includes the **errorInfo** node since individual charges could have their own errors.

```
<MRICMChargesResponse>
  <!-- ID of the processing batch in MRI -->
  <transactionID>a common identifier of this
  transaction</transactionID>
  <!-- if there are errors during accepting of charges payload,
  validation or processing -->
  <errors>
    <error>
      <item>number of the charge failed, does not exist
       for overall batch processing failures</item>
      <description>user-readable description of the
       error</description>
    </error>
    ...
  </errors>
</MRICMChargesResponse>
```

Lease Read Interface (LEAS)

This method provides the ability to read commercial lease information from the !{Lease}s table (LEAS) in an MRI database. This method is available for all installations that include a license for the Commercial Management application. Only specified standard fields, plus additional custom fields, will be included in the response. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRICMLeaseRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRICMLeaseRead>
```

Overlay 2—A lease for the specified building ID.

```
<IRESMRICMLeaseRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <BLDGID>string</BLDGID>                       Required
</IRESMRICMLeaseRead>
```

Overlay 3—A single row for the specified building and lease ID.

```
<IRESMRICMLeaseRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <BLDGID>string</BLDGID>                       Required
  <LEASID>string</LEASID>                       Required
</IRESMRICMLeaseRead>
```

Overlay 4—All rows updated since a specified last update date.

```
<IRESMRICMLeaseRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LASTDATE>datetime</LASTDATE>                 Required
</IRESMRICMLeaseRead>
```

returnedInfo (API response)

```
<IRESMRICMLeaseReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>999</rowsReturned>
  <LEAS>
    <BLDGID>string</BLDGID>
    <LEASID>string</LEASID>
    <SUITID>string</SUITID>
    <OCCPNAME>string</OCCPNAME>
    <DBA>string</DBA>
    <CONTNAME>string</CONTNAME>
    <PHONENO1>string</PHONENO1>
    <PHONENO2>string</PHONENO2>
    <NAME>string</NAME>
    <ADDRESS>string</ADDRESS>
    <ADDRESS2>string</ADDRESS2>
    <CITY>string</CITY>
    <STATE>string</STATE>
    <ZIPCODE>string</ZIPCODE>
    <ATTENT>string</ATTENT>
    <TENTID>string</TENTID>
    <TTYPID>string</TTYPID>
    <OCCPSTAT>string</OCCPSTAT>
    <EXECDATE<datetime</EXECDATE>
    <RENTSTRT<datetime</RENTSTRT>
    <OCCUPNCY<datetime</OCCUPNCY>
    <BEGINDATE<datetime</BEGINDATE>
    <EXPIR<datetime</EXPIR>
    <VACATE<datetime</VACATE>
    <CANCDATE<datetime</CANCDATE>
    <VENDID>string</VENDID>
    <COUNTY>string</COUNTY>
```

```
    <MOCCPID>string</MOCCPID>
    <GENERATION>numeric</GENERATION>
    <GENCODE>string</GENCODE>
    <ADDLSPACE>string</ADDLSPACE>
    <STOPBILLDATE<datetime</STOPBILLDATE>
    <CURRCODE>string</CURRCODE>
    <DEPARTMENT>string</DEPARTMENT>
    <EMAILBILL>string</EMAILBILL>
    <LASTDATE<datetime</LASTDATE>
    <USERID>string</USERID>
    …{additional custom field tags}
  </LEAS>
  …
</IRESMRICMLeaseReadResponse>
```

errorInfo (error response)

```
<IRESMRICMLeaseReadErrorMessage>
  <error>
    <message>
      <IRESMRICMLeaseReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRICMLeaseReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRICMLeaseReadErrorMessage>
```

Expense Pool Accounts Read Interface (IRESMRICMExpensePoolRead)

This method provides the ability to read commercial building expense pool accounts in an MRI database. This method is available for all installations that include a license for the Commercial Management application. This interface does not provide support for additional custom fields. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRICMExpensePoolRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRICMExpensePoolRead>
```

Overlay 2—All expense pools for a specified building ID.

```
<IRESMRICMExpensePoolRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <BLDGID>string</BLDGID>                       Required
</IRESMRICMExpensePoolRead>
```

Overlay 3—A single expense pool for a specified building ID, income category and expense pool ID.

```
<IRESMRICMExpensePoolRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <BLDGID>string</BLDGID>                       Required
  <INCCAT>string</INCCAT>                       Required
  <POOLID>string</POOLID>                       Required
</IRESMRICMExpensePoolRead>
```

returnedInfo (API response)

```
<IRESMRICMExpensePoolReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>999</rowsReturned>
  <CMPOOL>
    <BLDGID>string</BLDGID>
    <INCCAT>string<INCCAT>
    <POOLID>string</POOLID>
    <accountList>
      <ACCTNUM>string</ACCTNUM>
    </accountList>
  <CMPOOL>
  …
</IRESMRICMExpensePoolReadResponse>
```

errorInfo (error response)

```
<IRESMRICMExpensePoolReadErrorMessage>
  <error>
    <message>
      <IRESMRICMExpensePoolReadError>
        <error>
          <BLDGID>string</BLDGID>               Where available
          <INCCAT>string</INCCAT>               Where available
          <POOLID>string</POOLID>               Where available
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRICMExpensePoolReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRICMExpensePoolReadErrorMessage>
```

General Ledger APIs

Account Update Interface (IRESMRIGLAccountUpdate)

This method provides the ability to update the financial chart of accounts information in the GL Chart of Accounts table (GACC) of an MRI database. This method is available for all installations as the GACC table is a base table and is not associated with a specific application. If a record already exists for a provided ACCTNUM field, the information in the record will be updated with the information in the request. If a record does not already exist for a provided ACCTNUM field, a new record will be inserted into the table. The request will **not** be treated as a batch. As a result, all valid rows will be processed, and the database will be updated with errors returned for unsuccessful items. This method will support additional custom fields and the specified data validations. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

```
<IRESMRIGLAccountUpdate>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <GACC>
    <ACCTNUM>string</ACCTNUM>                   Required
    <ACCTNAME>string</ACCTNAME>                 Optional
    <TYPE>string</TYPE>                         Req'd for Insert
    <M_1099ACCT>string</M_1099ACCT>             Optional
    <DPRSTR>string</DPRSTR>                     Optional
    <PEXCHTYPE>string</PEXCHTYPE>               Optional
    <OWNERTAX>string</OWNERTAX>                 Optional
    <SUBWITH>string</SUBWITH>                   Optional
    <ACCTBASIS>string</ACCTBASIS>               Optional
    <LEGALACCT>string</LEGALACCT>               Optional
    <LASTDATE>datetime</LASTDATE>               Blocked
    <USERID>string</USERID>                     Blocked
    …{additional custom field tags}             Optional
  </GACC>
  …
</IRESMRIGLAccountUpdate>
```

Note

- M_1099ACCT will default to **N** if not provided.
- PEXCHTYPE will only be accepted if multiple currency is enabled.
- SUBWITH will default to **Y** if not provided.
- LEGALACCT will default to **Y** if not provided.
- DEPRECIATE will default to **N**.

- LASTDATE and USERID are updated using the current time and authenticated user ID.
- Additional custom field tags are mapped to custom fields in the GACC table by the field name. No additional validation will be performed beyond the foreign key validation with default values being set from the MRIFIELD definition where provided.

returnedInfo (API response)

```
<IRESMRIGLAccountUpdateResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsUpdated>numeric</rowsUpdated>
  <rowsInserted>numeric</rowsInserted>
  <rowsValidated>numeric</rowsValidated>
  <rowsInError>numeric</rowsInError>
</IRESMRIGLAccountUpdateResponse>
```

errorInfo (error response)

```
<IRESMRIGLAccountUpdateErrorMessage>
  <error>
    <message>
      <IRESMRIGLAccountUpdateError>
        <error>
          <ACCTNUM>string</ACCTNUM>              Where available
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLAccountUpdateError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLAccountUpdateErrorMessage>
```

Department Update Interface (IRESMRIGLDepartmentUpdate)

This method provides the ability to update the financial department information in the !{Departments} table (GDEP) of an MRI database. This method is available for all installations that include a license for the General Ledger application. If a record already exists for a provided department, the information in the record will be updated with the information in the request. If a record does not already exist for a provided department, a new record will be inserted into the table. The request will **not** be treated as a batch. As a result, all valid rows will be processed and the database will be updated with errors returned for unsuccessful items. This method will provide support for additional custom fields, foreign key validation, and defaulting based on the definition in the MRIField table. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

```
<IRESMRIGLDepartmentUpdate>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <GDEP>
    <DEPARTMENT>string</DEPARTMENT>             Required
    <DESCRPN>string</DESCRPN>                   Req'd for Insert
    <LASTDATE>datetime</LASTDATE>               Blocked
    <USERID>string</USERID>                     Blocked
    …{additional custom field tags}             Optional
  </GDEP>
  …
</IRESMRIGLDepartmentUpdate>
```

Note

- LASTDATE and USERID are updated using the current date and time and user ID.
- Additional custom field tags are mapped to custom fields in the GDEP table by the field name. No additional validation will be performed beyond the foreign key validation with default values being set from the MRIFIELD definition where provided.

returnedInfo (API response)

```
<IRESMRIGLDepartmentUpdateResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsUpdated>numeric</rowsUpdated>
  <rowsInserted>numeric</rowsInserted>
  <rowsValidated>numeric</rowsValidated>
  <rowsInError>numeric</rowsInError>
</IRESMRIGLDepartmentUpdateResponse>
```

errorInfo (error response)

```
<IRESMRIGLDepartmentUpdateErrorMessage>
  <error>
    <message>
      <IRESMRIGLDepartmentUpdateError>
        <error>
          <DEPARTMENT>string</DEPARTMENT>        Where available
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLDepartmentUpdateError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLDepartmentUpdateErrorMessage>
```

Job Code Update Interface (IRESMRIGLJobCodeUpdate)

This method provides the ability to update the job code information in the !{Job Code}s table (GJOB) of an MRI database. This method is available for all installations with a General Ledger application license. If a record already exists for a provided job code, the information in the record will be updated with the information in the request. If a record does not already exist for a provided job code, a new record will be inserted into the table. The request will **not** be treated as a batch. As a result, all valid rows will be processed, and the database will be updated with errors returned for unsuccessful items. This method will provide support for additional custom fields, foreign key validation, and defaulting based on the definition in the MRIField table. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

```
<IRESMRIGLJobCodeUpdate>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <GJOB>
    <JOBCODE>string</JOBCODE>                   Required
    <DESCRPTN>string</DESCRPTN>                 Req'd for Insert
    <JOBTYPE>string</JOBTYPE>                   Req'd for Insert
    <JOBBUDGET>string</JOBBUDGET>               Optional
    <JOBSRTDT>datetime</JOBSRTDT>               Req'd for Insert
    <JINACTDT>datetime</JINACTDT>               Optional
    <JACTIVE>string</JACTIVE>                   Req'd for Insert
    <SQFEET>numeric</SQFEET>                    Optional
    <BLDGID>string</BLDGID>                     Optional
    <LEASID>string</LEASID>                     Optional
    <REVENH>string</REVENH>                     Optional
    <CAPRECUR>string</CAPRECUR>                 Optional
    <PROJAMT>numeric</PROJAMT>                  Optional
    <CHNGAMT>numeric<CHNGAMT>                   Optional
```

```
    <ESCYRS>numeric</ESCYRS>                    Optional
    <ACCTNUM>string</ACCTNUM>                   Optional
    <INSVCDT>datetime</INSVCDT>                 Optional
    <ASSETID>string</ASSETID>                   Optional
    <LIFE>numeric</LIFE>                        Optional
    <MANAGER>string</MANAGER>                   Optional
    <JOBCOST>string</JOBCOST>                   Optional
    <JC_COSTLIST>string</COSTLIST>              Optional
    <PCTGCOMP>numeric</PCTRCOMP>                Optional
    <RMPROPID>string</RMPROPID>                 Optional
    <RMBLDGID>string</RMBLDGID>                 Optional
    <RMUNIT>string</RMUNIT>                     Optional
    <RMSQFEET>numeric</RMSQFEET>                Optional
    <SUITID>string</SUITID>                     Optional
    <JIMPCDT>datetime</JIMPCDT>                 Optional
    <COMMENT>string</COMMENT>                   Optional
    <LASTDATE>datetime</LASTDATE>               Blocked
    <USERID>string</USERID>                     Blocked
    …{additional custom field tags}             Optional
  </GJOB>
  …
</IRESMRIGLJobCodeUpdate>
```

Note

- JOBCOST will be set to **N** unless specified.
- LASTDATE and USERID are updated using the current date and time and user ID.
- Additional custom field tags are mapped to custom fields in the GJOB table by the field name. No additional validation will be performed beyond the foreign key validation with default values being set from the MRIFIELD definition where provided.

returnedInfo (API response)

```
<IRESMRIGLJobCodeUpdateResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsUpdated>numeric</rowsUpdated>
  <rowsInserted>numeric</rowsInserted>
  <rowsValidated>numeric</rowsValidated>
  <rowsInError>numeric</rowsInError>
</IRESMRIGLJobCodeUpdateResponse>
```

errorInfo (error response)

```
<IRESMRIGLJobCodeUpdateErrorMessage>
  <error>
    <message>
      <IRESMRIGLJobCodeUpdateError>
        <error>
          <JOBCODE>string</JOBCODE>             Where available
          <message>string</message>
```

```
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLJobCodeUpdateError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLJobCodeUpdateErrorMessage>
```

Journal Entry Read Interface (IRESMRIGLJournalRead)

This method provides the ability to read financial journal entry information from the Current Journal Entries table (JOURNAL) and the GL Closed Detail table (GHIS) in an MRI database. This interface is available for all installations that include a license for the General Ledger application. Multiple request structures will be supported as described below. This interface is processed through the use of generic read logic and will include additional custom fields in the response. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All journal entries for a specified entity ID, GL period, account number, and accounting basis.

```
<IRESMRIGLJournalRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <ENTITYID>string</ENTITYID>                   Required
  <PERIOD>string</PERIOD>                       Required
  <ACCTNUM>string</ACCTNUM>                     Required
  <BASIS>string</BASIS>                         Required
</IRESMRIGLJournalRead>
```

Overlay 2—All journal entries for a specified GL period, reference, source, and site ID.

```
<IRESMRIGLJournalRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <PERIOD>string</PERIOD>                       Required
  <REF>string</REF>                             Required
  <SOURCE>string</SOURCE>                       Required
  <SITEID>string</SITEID>                       Required
</IRESMRIGLJournalRead>
```

Overlay 3—All journal entries for a specified entity ID, GL period, and accounting basis.

```
<IRESMRIGLJournalRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <ENTITYID>string</ENTITYID>                   Required
  <PERIOD>string</PERIOD>                       Required
  <BASIS>string</BASIS>                         Required
</IRESMRIGLJournalRead>
```

Overlay 4—All journal entries for a specified reference.

```
<IRESMRIGLJournalRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <REF>string</REF>                             Required
</IRESMRIGLJournalRead>
```

returnedInfo (API response)

```
<IRESMRIGLJournalReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>9999</rowsReturned>
  <JOURNAL>
    <PERIOD>string</PERIOD>
    <REF>string</REF>
    <SOURCE>string</SOURCE>
    <SITEID>string</SITEID>
    <ITEM>numeric</ITEM>
    <ENTITYID>string</ENTITYID>
    <ACCTNUM>string</ACCTNUM>
    <DEPARTMENT>string</DEPARTMENT>
    <JOBCODE>string</JOBCODE>
    <AMT>numeric</AMT>
    <DESCRPN>string</DESCRPN>
    <ENTRDATE>datetime</ENTRDATE>
    <BASIS>string</BASIS>
    <AUDITFLAG>string</AUDITFLAG>
    <JC_PHASECODE>string</JC_PHASECODE>
    <JC_COSTLIST>string</JC_COSTLIST>
    <JC_COSTCODE>string</JC_COSTCODE>
    <CATEGORY>string</CATEGORY>
    <ADDLDESC>string</ADDLDESC>
    <CURRCODE>string</CURRCODE>
…{additional custom field tags}
  </JOURNAL>
</IRESMRIGLJournalReadResponse>
```

Note

CURRCODE will be set based on the currency code of the entity or database if available.

errorInfo (error response)

```
<IRESMRIGLJournalReadErrorMessage>
  <error>
    <message>
      <IRESMRIGLJournalReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLJournalReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLJournalReadErrorMessage>
```

Journal Entry Marked Read Interface (IRESMRIGLJournalMarkedRead)

This method provides the ability to read unmarked financial journal entry information from the JOURNAL and GHIS tables in an MRI database. This method is available for all installations that include a license for the General Ledger application. Unmarked fields will be defined as any record where the **INTERFACEID** field is null. An additional optional filter will be available for limiting reads to a single entity ID. This method will include additional custom fields in the response. Marking will occur through the update of an additional field on the JOURNAL and GHIS tables named **INTERFACEID**. The interface ID will be a system-generated value and will be included in the response. Marking will only occur if the **markRows** tag is included. Otherwise, all unmarked rows will be returned, but marking will not be done and the interface ID will not be generated. If a **clientBatchMarker** tag is included in the request, the value of this tag will be used to update the JOURNAL or GHIS table. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

Note

When using the Journal Entry Marked Read Interface, performance can be improved through the creation of the following additional indexes.

- **GL Closed Detail Table (GHIS)**—Add a non-unique index on INTERFACEID.
- **Current Journal Entries Table (JOURNAL)**—Add a non-unique index on INTERFACEID.

These indexes are not included with the standard installation due to the likelihood that they would slow the saving of journal entries and payments through the standard user interface.

commandInfo (request)

Overlay 1—All journal entries for a specified entity ID that are unmarked.

```
<IRESMRIGLJournalMarkedRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <markRows/>                                   Optional
  <clientBatchMarker>string</clientBatchMarker> Optional
  <ENTITYID>string</ENTITYID>                   Required
</IRESMRIGLJournalMarkedRead>
```

Overlay 2—All unmarked journal entries.

```
<IRESMRIGLJournalMarkedRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <markRows/>                                   Optional
  <clientBatchMarker>string</clientBatchMarker> Optional
  <all-rows>Y</all-rows>                        Optional
</IRESMRIGLJournalMarkedRead>
```

Note

The presence of the **markRows** tag determines if records are to be marked.

returnedInfo (API response)

```
<IRESMRIGLJournalMarkedReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <interfaceId>12345678</interfaceId>
  <rowsReturned>9999</rowsReturned>
  <rowsMarked>9999</rowsMarked>
  <JOURNAL>
    <PERIOD>string</PERIOD>
    <REF>string</REF>
    <SOURCE>string</SOURCE>
    <SITEID>string</SITEID>
    <ITEM>numeric</ITEM>
    <ENTITYID>string</ENTITYID>
    <ACCTNUM>string</ACCTNUM>
    <DEPARTMENT>string</DEPARTMENT>
    <JOBCODE>string</JOBCODE>
    <AMT>numeric</AMT>
    <DESCRPN>string</DESCRPN>
    <ENTRDATE>datetime</ENTRDATE>
    <REVERSAL>string</REVERSAL>
    <BASIS>string</BASIS>
    <AUDITFLAG>string</AUDITFLAG>
    <JC_PHASECODE>string</JC_PHASECODE>
    <JC_COSTLIST>string</JC_COSTLIST>
    <JC_COSTCODE>string</JC_CODECODE>
    <CATEGORY>string</CATEGORY>
    <ADDLDESC>string</ADDLDESC>
    <CURRCODE>string</CURRCODE>
…{additional custom field tags}
```

```
  </JOURNAL>
  …
</IRESMRIGLJournalMarkedReadResponse>
```

Note

The CURRCODE column will be set based on the currency code of the entity or database, if available.

errorInfo (error response)

```
<IRESMRIGLJournalMarkedReadErrorMessage>
  <error>
    <message>
      <IRESMRIGLJournalMarkedReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLJournalMarkedReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLJournalMarkedReadErrorMessage>
```

Note

To prevent modification of synchronized data, changes must be made to both the Windows and Web user interface for journal entry. The changes will consist of checking the **INTERFACEID** field. If the value is not null, changes should not be allowed to the journal entry.

Journal Entry Create Interface (IRESMRIGLJournalCreate)

This method provides the ability to create new journal entries in the JOURNAL and GHIS tables in an MRI database, but does not support the ability to update existing journal entries. This method is available for all installations that include a license for the General Ledger application.

If the request includes entries for closed periods, an error will be returned unless the appropriate authorization override (prior period or year) flags have been included. If the appropriate authorization is included, entries will be processed without error. Future period controls will also be enforced according to the setting for the supplied entity. If the **Require JEs to Balance** check box is selected in the management options for General Ledger, all journal entries included in a request will be validated to ensure they balance. If entries are not in balance, an error will be returned. All entries created through this method will be considered posted. Journal entry numbering will not be supported through this method as it is not supported in the current Web interface. In addition, this method will not support the creation of reversing entries. The request will be

treated as a batch. As a result, any error will result in rejection of the whole request.

The **INTERFACEID** field on the JOURNAL or GHIS table will be updated with a system-generated value and that value will be returned as part of the response. If the request contains the optional **clientBatchMarker** tag, the INTFMARKER field in JOURNAL or GHIS table will be updated with that value.

This method will provide support for additional custom fields, foreign key validation and defaulting based on the definition in the MRIField table. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

```
<IRESMRIGLJournalCreate>
  <interfaceVersion>1.0</interfaceVersion>       Optional
  <authorizePriorPeriod>Y</authorizePriorPeriod> Optional
  <authorizePriorYear>Y<authorizePriorYear>      Optional
  <clientBatchMarker>string</clientBatchMarker> Optional
  <controlTotal>numeric</controlTotal>           Optional
  <JOURNAL>
    <PERIOD>string</PERIOD>                      Required
    <REF>string</REF>                            Blocked
    <SOURCE>string</SOURCE>                      Required
    <SITEID>string</SITEID>                      Blocked
    <ITEM>numeric</ITEM>                         Blocked
    <ENTITYID>string</ENTITYID>                  Required
    <ACCTNUM>string</ACCTNUM>                    Required
    <DEPARTMENT>string</DEPARTMENT>              Optional
    <JOBCODE>string</JOBCODE>                    Optional
    <AMT>numeric</AMT>                           Required
    <DESCRPN>string</DESCRPN>                    Required
    <ENTRDATE>datetime</ENTRDATE>                Optional
    <REVERSAL>string</REVERSAL>                  Blocked
    <STATUS>string</STATUS>                      Blocked
    <OEXCHREF>string</OEXCHREF>                  Blocked
    <BASIS>string</BASIS>                        Required
    <OCURRCODE>string</OCURRCODE>                Blocked
    <OAMT>numeric</OAMT>                         Blocked
    <REVREF>string</REVREF>                      Blocked
    <INTENTENTRY>string</INTENTENTRY>            Blocked
    <INTENTTYPE>string</ENTENTTYPE>              Blocked
    <CJEGRPID>string</CJEGRPID>                  Blocked
    <CJEID>string</CJEID>                        Blocked
    <DESTITEM>numeric</DESTITEM>                 Blocked
    <AUDITFLAG>string</AUDITFLAG>                Optional
    <REVENTRY>string</REVENTRY>                  Blocked
    <REVPRD>string</REVPRD>                      Blocked
    <REVITEM>numeric</REVITEM>                   Blocked
    <OWNERTAX>string</OWNERTAX>                  Blocked
    <OWNPCTCALC>numeric</OWNPCTCALC>             Blocked
```

```
    <JC_PHASECODE>string</JC_PHASECODE>          Optional
    <JC_COSTLIST>string</JC_COSTLIST>            Optional
    <JC_COSTCODE>string</JC_COSTCODE>            Optional
    <CATEGORY>string</CATEGORY>                  Optional
    <BRSTATUS>string</BRSTATUS>                  Blocked
    <OWNTBL>string</OWNTBL>                      Blocked
    <OWNYEAR>string</OWNYEAR>                    Blocked
    <OWNNUM>numeric</OWNNUM>                     Blocked
    <CTRYTBL>string</CTRYTBL>                    Blocked
    <CTRYYEAR>string</CTRYYEAR>                  Blocked
    <BANKRECID>string</BANKRECID>                Blocked
    <FASALEREF>string</FASALEREF>                Blocked
    <PACKADJ>string</PACKADJ>                    Blocked
    <PACKAGEID>string</PACKAGEID>                Blocked
    <PACKPROC>string</PACKPROC>                  Blocked
    <PACKINIT>string</PACKINIT>                  Blocked
    <ADDLDESC>string</ADDLDESC>                  Optional
    <LASTDATE>datetime</LASTDATE>                Blocked
    <USERID>string</USERID>                      Blocked
    <CURRCODE>string</CURRCODE>                  Optional
…{additional custom field tags}                  Optional
  </JOURNAL>
  …
</IRESMRIGLJournalCreate>
```

Note

- REF is blocked and automatically numbered regardless of the database setting.
- SITEID that is assigned to the Web Services user. Otherwise, default to @.
- ITEM is automatically numbered.
- DEPARTMENT defaults to @ if not provided.
- ENTRDATE defaults to the current date if not provided.
- ENTRDATE must be entered in ISO-8601 format, such as 2011-10-18 for October 18, 2011.
- BASIS defaults to the Basis setting on the GL !{Entity} table (ENTITY) for the provided entity if not specifically provided.
- Additional custom field tags are mapped to custom fields in the JOURNAL table by the field name. No additional validation will be performed beyond the foreign key validation with default values being set from the MRIField table where provided.

returnedInfo (API response)

```
<IRESMRIGLJournalCreateResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <clientBatchMarker>string</clientBatchMarker>
  <rowsInserted>numeric</rowsInserted>
  <rowsValidated>numeric</rowsValidated>
  <rowsInError>numeric</rowsInError>
```

```
  <interfaceId>99999999</interfaceId>
  <references>
    <reference>
      <REF>string</REF>
      <requestSequenceNumber>numeric</requestSequenceNumber>
      …
    </reference>
    …
  </references>
</IRESMRIGLJournalCreateResponse>
```

errorInfo (error response)

```
<IRESMRIGLJournalCreateErrorMessage>
  <error>
    <message>
      <IRESMRIGLJournalCreateError>
        <error>
          <PERIOD>string</PERIOD>                Where available
          <SOURCE>string</SOURCE>                Where available
          <SITEID>string</SITEID>                Where available
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLJournalCreateError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLJournalCreateErrorMessage>
```

GL Summary Read Interface (IRESMRIGLSummaryRead)

This method provides the ability to read financial period activity and balance summary information from the GL Summary table (GLSUM) in an MRI database. This method is available for all installations that include a license for the General Ledger application. Multiple request structures are supported as described below, but no support will be available for **BALFOR** as a parameter. No support is provided for additional custom fields. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All journal entries for a specified entity ID, GL period, account number and accounting basis.

```
<IRESMRIGLSummaryRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <ENTITYID />                                  Required
```

```
  <PERIOD />                                    Required
  <BASIS />                                     Optional
  <DEPARTMENT />                                Optional
  <ACCTNUM />                                   Optional
</IRESMRIGLSummaryRead>
```

Note

DEPARTMENT will default to @ if not provided.

returnedInfo (API response)

```
<IRESMRIGLSummaryReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>9999</rowsReturned>
  <GLSUM>
    <PERIOD>string</PERIOD>
    <ENTITYID>string</ENTITYID>
    <BASIS>string</BASIS>
    <DEPARTMENT>string</DEPARTMENT>
    <ACCTNUM>string</ACCTNUM>
    <ACTIVITY>numeric</ACTIVITY>
   </GLSUM>
  …
</IRESMRIGLSummaryReadResponse>
```

errorInfo (error response)

```
<IRESMRIGLSummaryReadErrorMessage>
  <error>
    <message>
      <IRESMRIGLSummaryReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLSummaryReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLSummaryReadErrorMessage>
```

Budget Read Interface (IRESMRIGLBudgetRead)

This method provides the ability to read financial budget information from the Budgets table (BUDGETS) in an MRI database. This method is available for all installations that include a license for the General Ledger application. This interface does not provide support for additional custom fields. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All budgets for the specified budget type, year ending period, entity ID, accounting basis and department.

```
<IRESMRIGLBudgetRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <BUDTYPE>string</BUDTYPE>                     Required
  <YEARENDINGPD>string</YEARENDINGPD>           Required
  <ENTITYID>string</ENTITYID>                   Optional
  <BASIS>string</BASIS>                         Optional
  <DEPARTMENT>string</DEPARTMENT>               Optional
</IRESMRIGLBudgetRead>
```

Note

DEPARTMENT will default to @ if not provided.

returnedInfo (API response)

```
<IRESMRIGLBudgetReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>9999</rowsReturned>
  <BUDGET12>
    <BUDTYPE>string</BUDTYPE>
    <ENTITID>string</ENTITYID>
    <DEPARTMENT>string</DEPARTMENT>
    <BASIS>string</BASIS>
    <YEARENDINGPD>string</YEARENDINGPD>
    <CURRCODE>string</CURRCODE>
    <budgetLine>
      <ACCTNUM>string</ACCTNUM>
      <ACTIVITY-01>numeric</ACTIVITY-01>
      <ACTIVITY-02>numeric</ACTIVITY-02>
      <ACTIVITY-03>numeric</ACTIVITY-03>
      <ACTIVITY-04>numeric</ACTIVITY-04>
      <ACTIVITY-05>numeric</ACTIVITY-05>
      <ACTIVITY-06>numeric</ACTIVITY-06>
      <ACTIVITY-07>numeric</ACTIVITY-07>
      <ACTIVITY-08>numeric</ACTIVITY-08>
      <ACTIVITY-09>numeric</ACTIVITY-09>
      <ACTIVITY-10>numeric</ACTIVITY-10>
      <ACTIVITY-11>numeric</ACTIVITY-11>
      <ACTIVITY-12>numeric</ACTIVITY-12>
      <budgetNote>
        <NOTETYPE>string</NOTETYPE>
        <BEGPD>string</BEGPD>
        <ENDPD>string</ENDPD>
        <NOTETEXT>string</NOTETEXT>
        <LASTDATE>datetime</LASTDATE>
        <USERID>string</USERID>
      </budgetNote>
      ...
    </budgetLine>
    …
```

```
  </BUDGET12>
  …
</IRESMRIGLBudgetReadResponse>
```

errorInfo (error response)

```
<IRESMRIGLBudgetReadErrorMessage>
  <error>
    <message>
      <IRESMRIGLBudgetReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLBudgetReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLBudgetReadErrorMessage>
```

Budget Update Interface (IRESMRIGLBudgetUpdate)

This method provides the ability to update financial budget information in the BUDGETS table in an MRI database. This method is available for all installations that include a license for the General Ledger application. If any locked budget records exist for the specified BUDTYPE, ENTITYID, DEPARTMENT, BASIS and YEARENDINGPD (LOCKED=Y), an error will be returned unless the appropriate authorization flag has been included. All records for the existing BUDTYPE, ENTITY, BASIS, DEPARTMENT and YEARENDINGPD will be deleted prior to updating with the request data. Periods will be calculated with period 12 of the budget year equaling the YEARENDINGPD. The request will be treated as a batch. As a result, any error in the request will cause the rejection of the entire request. This interface does not provide support for additional custom fields. The budget update logic will convert the spreadsheet format interface to the normalized structure of the BUDGETS table. The parameter sets below define how standard fields will be processed.

commandInfo (request)

```
<IRESMRIGLBudgetUpdate>
  <interfaceVersion>1.0</interfaceVersion>
  <authorizeBudgetLockOverride>Y</authorizeBudgetLockOverride>
  <BUDGET12>
    <BUDTYPE>string</BUDTYPE>                    Required
    <ENTITYID>string</ENTITYID>                  Required
    <DEPARTMENT>string</DEPARTMENT>              Optional
    <BASIS>string</BASIS>                        Required
    <YEARENDINGPD>string</YEARENDINGPD>          Required
    <LOCKED>string</LOCKED>                      Optional
```

```
    <LASTDATE>datetime</LASTDATE>                Blocked
    <USERID>string</USERID>                      Blocked
    <budgetLine>
      <ACCTNUM>string</ACCTNUM>                  Required
      <ACTIVITY-01>numeric</ACTIVITY-01>         Optional
      <ACTIVITY-02>numeric</ACTIVITY-02>         Optional
      <ACTIVITY-03>numeric</ACTIVITY-03>         Optional
      <ACTIVITY-04>numeric</ACTIVITY-04>         Optional
      <ACTIVITY-05>numeric</ACTIVITY-05>         Optional
      <ACTIVITY-06>numeric</ACTIVITY-06>         Optional
      <ACTIVITY-07>numeric</ACTIVITY-07>         Optional
      <ACTIVITY-08>numeric</ACTIVITY-08>         Optional
      <ACTIVITY-09>numeric</ACTIVITY-09>         Optional
      <ACTIVITY-10>numeric</ACTIVITY-10>         Optional
      <ACTIVITY-11>numeric</ACTIVITY-11>         Optional
      <ACTIVITY-12>numeric</ACTIVITY-12>         Optional
      <budgetNote>                               Optional
        <NOTETYPE>string</NOTETYPE>              Optional
        <BEGPD>string</BEDPD>                    Required
        <ENDPD>string</ENDPD>                    Required
        <NOTETEXT>string</NOTETEXT>              Required
        <LASTDATE>datetime</LASTDATE>            Blocked
        <USERID>string</USERID>                  Blocked
      </budgetNote>                              Optional
      …
    </budgetLine>
    …
  </BUDGET12>
  …
</IRESMRIGLBudgetUpdate>
```

Note

- DEPARTMENT will default to @ if not provided.
- LOCKED will default to N if not provided.
- NOTETYPE will default to B if not provided.

returnedInfo (API response)

```
<IRESMRIGLBudgetUpdateResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsInserted>numeric</rowsInserted>
  <rowsValidated>numeric</rowsValidated>
  <rowsInError>numeric</rowsInError>
</IRESMRIGLBudgetUpdateResponse>
```

errorInfo (error response)

```
<IRESMRIGLBudgetUpdateErrorMessage>
  <error>
    <message>
      <IRESMRIGLBudgetUpdateError>
        <error>
```

```
          <BUDGTYPE>string</BUDTYPE>              Where available
          <ENTITYID>string</ENTITYID>             Where available
          <DEPARTMENT>string</DEPARTMENT>         Where available
          <BASIS>string</BASIS>                   Where available
          <YEARENDINGPD>string</YEARENDINGPD>     Where available
          <ACCTNUM>string</ACCTNUM>               Where available
          <NOTETYPE>string</NOTETYPE>             Where available
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLBudgetUpdateError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLBudgetUpdateErrorMessage>
```

Budget Type Read Interface (IRESMRIGLBudgetTypeRead)

This method provides the ability to read financial budget type information from the Budget Types table (GBTY) in an MRI database. This method is available for all installations that include a license for the General Ledger application. All fields from the GBTY table, plus additional custom fields, will be returned in the response. This interface does not provide support for additional custom fields. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRIGLBudgetTypeRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRIGLBudgetTypeRead>
```

Overlay 2—A single row for the specified budget type.

```
<IRESMRIGLBudgetTypeRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <BUDTYPE>string</BUDTYPE>                     Required
</IRESMRIGLBudgetTypeRead>
```

Overlay 3—All rows updated since a specified last update date.

```
<IRESMRIGLBudgetTypeRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LASTDATE>datetime</LASTDATE>                 Required
</IRESMRIGLBudgetTypeRead>
```

returnedInfo (API response)

```
<IRESMRIGLBudgetTypeReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>9999</rowsReturned>
  <GBTY>
    <BUDTYPE>string</BUDTYPE>
    <DESCRPTN>string</DESCRPTN>
    <LASTDATE>datetime</LASTDATE>
    <USERID>string</USERID>
    …{additional custom field tags}
  </GBTY>
  …
</IRESMRIGLBudgetTypeReadResponse>
```

errorInfo (error response)

```
<IRESMRIGLBudgetTypeReadErrorMessage>
  <error>
    <message>
      <IRESMRIGLBudgetTypeReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLBudgetTypeReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLBudgetTypeReadErrorMessage>
```

Budget Type Update Interface (IRESMRIGLBudgetTypeUpdate)

This method provides the ability to update GL budget type information in the GBTY table in an MRI database. This method is available for all installations that include a license for the General Ledger application. If a record already exists for a provided BUDTYPE, the information in the record will be updated with that in the request. If a record does not already exist for a provided BUDTYPE, a new record will be inserted into the table. The request will **not** be treated as a batch. As a result, all valid rows will be processed and the database will be updated with errors returned for unsuccessful items. This method will provide support for additional custom fields, foreign key validation, and defaulting based on the definition in the MRIField table. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

```
<IRESMRIGLBudgetTypeUpdate>
  <GBTY>
    <BUDTYPE>string</BUDTYPE>                    Required
    <DESCRPTN>string</DESCRPTN>                  Req'd for Insert
…{additional custom field tags}                  Optional
  </GBTY>
</IRESMRIGLBudgetTypeUpdate>
```

Note

Additional custom field tags are mapped to custom fields in the GBTY table by the field name. No additional validation will be performed beyond the foreign key validation with default values being set from the MRIFIELD definition where provided.

returnedInfo (API response)

```
<IRESMRIGLBudgetTypeUpdateResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsUpdated>numeric</rowsUpdated>
  <rowsInserted>numeric</rowsInserted>
  <rowsValidated>numeric</rowsValidated>
  <rowsInError>numeric</rowsInError>
</IRESMRIGLBudgetTypeUpdateResponse>
```

errorInfo (error response)

```
<IRESMRIGLBudgetTypeUpdateErrorMessage>
  <error>
    <message>
      <IRESMRIGLBudgetTypeUpdateError>
        <error>
          <BUDTYPE>string</BUDTYPE>              Where available
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLBudgetTypeUpdateError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLBudgetTypeUpdateErrorMessage>
```

Ledger and Account Read Interface (IRESMRIGLLedgerAccountRead)

This method provides the ability to read financial ledger and account information from the Ledger Codes table (GLCD) and the GACC table in an MRI database. This method is available for all installations as the GLCD and GACC tables are base tables and are not associated with a specific application. Only the fields defined below, plus additional custom fields, will be returned in the response. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRIGLLedgerAccountRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRIGLLedgerAccountRead>
```

Overlay 2—All rows for a specified ledger code.

```
<IRESMRIGLLedgerAccountRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LEDGCODE>string</LEDGCODE>                   Required
</IRESMRIGLLedgerAccountRead>
```

Overlay 3—All ledger codes rows with all account (GACC) rows updated since a specified last update date.

```
<IRESMRIGLLedgerAccountRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LASTDATE>datetime</LASTDATE>                 Required
</IRESMRIGLLedgerAccountRead>
```

returnedInfo (API response)

```
<IRESMRIGLLedgerAccountReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>999</rowsReturned>
  <GLCD>
    <LEDGCODE>string</LEDGCODE>
    <DESCRPTN>string</DESCRPTN>
    <ACCTLGT>numeric</ACCTLGT>
    <ACCTDSP>string</ACCTDSP>
    <ACCTNUM>string</ACCTNUM>
    <ACCTMAJ>numeric</ACCTMAJ>
    <REARNACC>string</REARNACC>
    <REGNCODE>string</REGNCODE>
```

```
    <CODELEN>numeric</CODELEN>
    <LOCATIONS>string</LOCATION>
    …{additional custom fields}
    <GACC>
      <ACCTNUM>string</ACCTNUM>
      <ACCTNAME>string</ACCTNAME>
      <TYPE>string</TYPE>
      <M_1099ACCT>string</M_1099ACCT>
       <DPSTR>string</DPSTR>
       <PEXCHTYPE>string</PEXCHTYPE>
       <OWNERTAX>string</OWNERTAX>
       <SUBWITH>string</SUBWITH>
       <ACCTBASIS>string</ACCTBASIS>
       <LEGALACCT>string<L/LEGALACCT>
       …{Additional custom fields tags}
    </GACC>
    …
  </GLCD>
  …
</IRESMRIGLLedgerAccountReadResponse>
```

errorInfo (error response)

```
<IRESMRIGLLedgerAccountReadErrorMessage>
  <error>
    <message>
      <IRESMRIGLLedgerAccountReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLLedgerAccountReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLLedgerAccountReadErrorMessage>
```

Entity Read Interface (IRESMRIGLEntityRead)

This method provides the ability to read financial entity information from the ENTITY table in an MRI database. This method is available for all installations as the ENTITY table is a base table and is not associated with a specific application. Only the fields defined below, plus additional custom fields, will be returned in the response. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRIGLEntityRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRIGLEntityRead>
```

Overlay 2—A single row for the specified entity ID.

```
<IRESMRIGLEntityRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <ENTITYID>string</ENTITYID>                   Required
</IRESMRIGLEntityRead>
```

Overlay 3—All rows updated since a specified last update date.

```
<IRESMRIGLEntityRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LASTDATE>datetime</LASTDATE>                 Required
</IRESMRIGLEntityRead>
```

returnedInfo (API response)

```
<IRESMRIGLEntityReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>numeric</rowsReturned>
  <ENTITY>
    <ENTITYID>string</ENTITYID>
    <NAME>string</NAME>
    <ADDR1>string</ADDR1>
    <ADDR2>string</ADDR2>
    <ADDR3>string</ADDR3>
    <CITY>string</CITY>
    <STATE>string</STATE>
    <ZIPCODE>string</ZIPCODE>
    <COUNTRY>string</COUNTRY>
    <PHONE>string</PHONE>
    <ATAXACCT>string</ATAXACCT>
    <ATAXCASH>string</ATAXCASH>
    <BASIS>string</BASIS>
    <CASHTYPE>string</CASHTYPE>
    <CHECKLOCATIONID>string</CHECKLOCATIONID>
    <CURPED>string</CURPED>
    <CURRCODE>string</CURRCODE>
    <ENTTYPE>string</ENTTYPE>
    <INVCSTAT>string</INVCSTAT>
    <LEDGCODE>string</LEDGCODE>
    <MAXAPOPEN>numeric</MAXAPOPEN>
    <MAXOPEN>numeric</MAXOPEN>
    <OWNERID>string</OWNERID>
    <PROJID>string</PROJID>
    <STLINEBASIS>string</STLINEBASIS>
```

```
    <YEAREND>string</YEAREND>
    …{additional/custom fields}
  </ENTITY>
  …
</IRESMRIGLEntityReadResponse>
```

errorInfo (error response)

```
<IRESMRIGLEntityReadErrorMessage>
  <error>
    <message>
      <IRESMRIGLEntityReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLEntityReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLEntityReadErrorMessage>
```

Entity Cash Type Map Read Interface (IRESMRIGLEntityCashTypeMapRead)

This method provides the ability to read financial entity cash type information from the ENTITY table, the Cash/Bank Maps table (BMAP), and the !{Cash Type}s table (CTYP) in an MRI database. This method is available for all installations as the ENTITY and BMAP tables are base tables and are not associated with a specific application. Only the fields defined below, plus additional custom fields, will be returned in the response. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRIGLEntityCashTypeMapRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRIGLEntityCashTypeMapRead>
```

Overlay 2—All information for a specified entity ID.

```
<IRESMRIGLEntityCashTypeMapRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <ENTITYID>string</ENTITYID>                   Required
</IRESMRIGLEntityCashTypeMapRead>
```

returnedInfo (API response)

```
<IRESMRIGLEntityCashTypeMapReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>numeric</rowsReturned>
  <ENTITY>
    <ENTITYID>string</ENTITYID>
    <BMAP>
      <CASHTYPE>string</CASHTYPE>
      <CASHDESC>string</CASHDESC>
    </BMAP>
    …
  </ENTITY>
  …
</IRESMRIGLEntityCashTypeMapReadResponse>
```

Note

No support is provided for additional custom fields.

errorInfo (error response)

```
<IRESMRIGLEntityCashTypeMapReadErrorMessage>
  <error>
    <message>
      <IRESMRIGLEntityCashTypeMapReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLEntityCashTypeMapReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLEntityCashTypeMapReadErrorMessage>
```

Department Read Interface (IRESMRIGLDepartmentRead)

This method provides the ability to read department information from the GDEP table of an MRI database. This method is available for all installations that include a license for the General Ledger application. All fields from the GDEP table, plus additional custom fields, will be returned in the response. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRIGLDepartmentRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRIGLDepartmentRead>
```

Overlay 2—A single row for the specified department code.

```
<IRESMRIGLDepartmentRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <DEPARTMENT>string</DEPARTMENT>               Required
</IRESMRIGLDepartmentRead>
```

Overlay 2—All rows updated since a specified last update date.

```
<IRESMRIGLDepartmentRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LASTDATE>datetime</LASTDATE>                 Required
</IRESMRIGLDepartmentRead>
```

returnedInfo (API response)

```
<IRESMRIGLDepartmentReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>999</rowsReturned>
  <GDEP>
    <DEPARTMENT>string</DEPARTMENT>
    <DESCRPN>string</DESCRPN>
    <LASTDATE>datetime</LASTDATE>
    <USERID>string</USERID>
    …{additional custom field tags}
  </GDEP>
  …
</IRESMRIGLDepartmentReadResponse>
```

errorInfo (error response)

```
<IRESMRIGLDepartmentReadErrorMessage>
  <error>
    <message>
      <IRESMRIGLDepartmentReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLDepartmentReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLDepartmentReadErrorMessage>
```

Job Code Read Interface (IRESMRIGLJobCodeRead)

This method provides the ability to read job code information from the GJOB table in an MRI database. This method is available for all installations that include a license for the General Ledger application. All fields from the GJOB table, plus additional custom fields, will be returned in the response. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRIGLJobCodeRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRIGLJobCodeRead>
```

Overlay 2—A single row for the specified job code.

```
<IRESMRIGLJobCodeRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <JOBCODE>string</JOBCODE>                     Required
</IRESMRIGLJobCodeRead>
```

Overlay 3—All rows updated since a specified last update date.

```
<IRESMRIGLJobCodeRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LASTDATE>datetime</LASTDATE>                 Required
</IRESMRIGLJobCodeRead>
```

returnedInfo (API response)

```
<IRESMRIGLJobCodeReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>999</rowsReturned>
  <GJOB>
    <JOBCODE>string</JOBCODE>
    <DESCRPTN>string</DESCRPTN>
    <JOBTYPE>string</JOBTYPE>
    <JOBBUDGET>numeric</JOBBUDGET>
    <JOBSRTDT>datetime</JOBSRTDT>
    <JIMPCDT>datetime</JIMPCDT>
    <JINACTDT>datetime</JINACTDT>
    <JACTIVE>string</JACTIVE>
    <LASTDATE>datetime</LASTDATE>
    <USERID>string</USERID>
    <SQFEET>numeric</SQFEET>
    <BLDGID>string</BLDGID>
```

```
    <LEASID>string</LEASID>
    <REVENH>string</REVENH>
    <CAPRECUR>string</CAPRECUR>
    <PROJAMT>numeric</PROJAMT>
    <CHNGAMT>numeric</CHNGAMT>
    <ESCYRS>numeric</ESCYRS>
    <ACCTNUM>string</ACCTNUM>
    <INSVCDT>datetime</INSVCDT>
    <ASSETID>string</ASSETID>
    <LIFE>numeric</LIFE>
    <MANAGER>string</MANAGER>
    <JOBCOST>string<JOBCOST>
    <PCTGCOMP>numeric</PCTGCOMP>
    <RMPROPID>string</RMPROPID>
    <RMBLDGID>string</RMBLDGID>
    <RMUNIT>string</RMUNIT>
    <RMSQFEET>numeric</RMSQFEET>
    <COMMENT>string</COMMENT>
    <SUITID>string</SUITID>
    …{additional custom field tags}
  </GJOB>
  …
</IRESMRIGLJobCodeReadResponse>
```

errorInfo (error response)

```
<IRESMRIGLJobCodeReadErrorMessage>
  <error>
    <message>
      <IRESMRIGLJobCodeReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIGLJobCodeReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIGLJobCodeReadErrorMessage>
```

JobCost APIs

Interface Chart Interface (IRESMRIJCInterfaceRead)

This method provides the ability to read the JobCost interface chart in the Job Cost Interface Chart table (JC_MAP). This interface is available for all installations that include a license for the JobCost application. Multiple request structures will be supported as described below. This interface is processed through the use of generic read logic and includes additional custom fields in the response. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—Returns all rows in the JC_MAP table.

```
<IRESMRIJCInterfaceRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRIJCInterfaceRead>
```

Overlay 2—All rows updated since the last update date.

```
<IRESMRIJCInterfaceRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LASTDATE>datetime</LASTDATE>                 Required
</IRESMRIJCInterfaceRead>
```

returnedInfo (API response)

```
<IRESMRIJCInterfaceReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>numeric</rowsReturned>
  <JC_MAP>
    <JOBCODE>string</JOBCODE>
    <ENTITYID>string  </ENTITYID>
    <JC_PHASECODE>string</JC_PHASECODE>
    <JC_COSTLIST>string</JC_COSTLIST>
    <JC_COSTCODE>string</JC_COSTCODE>
    <DEPARTMENT>string</DEPARTMENT>
    <ACCTNUM>string</ACCTNUM>
    <ENTRDATE>datetime<ENTRDATE/>
    <LASTDATE>datetime</LASTDATE>
    <USERID>string</USERID>
    <ACCOUNTID>string<ACCOUNTID/>
  </JC_MAP>
</IRESMRIJCInterfaceReadResponse>
```

errorInfo (error response)

```
<IRESMRIJCInterfaceReadErrorMessage>
  <error>
    <message>
      <IRESMRIJCInterfaceReadError>
        <error>
          <message>String</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIJCInterfaceReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
</IRESMRIJCInterfaceReadErrorMessage>
```

Phase Read Interface (IRESMRIJCPhaseRead)

This method provides the ability to read phase information from the Job Cost Phases table (JC_PHASE) in an MRI database. This method is available for all installations that include a license for the JobCost application. All fields from the JC_PHASE table, plus additional custom fields, will be returned in the response. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRIJCPhaseRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRIJCPhaseRead>
```

Overlay 2—All phases for a specified job code.

```
<IRESMRIJCPhaseRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <JOBCODE>string</JOBCODE>                     Required
</IRESMRIJCPhaseRead>
```

Overlay 3—A single row for the specified job code and phase.

```
<IRESMRIJCPhaseRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <JOBCODE>string</JOBCODE>                     Required
  <JC_PHASECODE>string</JC_PHASECODE>           Required
</IRESMRIJCPhaseRead>
```

Overlay 4—All rows updated since a specified last update date.

```
<IRESMRIJCPhaseRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LASTDATE>datetime</LASTDATE>                 Required
</IRESMRIJCPhaseRead>
```

returnedInfo (API response)

```
<IRESMRIJCPhaseReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>999</rowsReturned>
  <JC_PHASE>
    <JOBCODE>string</JOBCODE>
    <JC_PHASECODE>string</JC_PHASECODE>
    <DESCRIPTION>string</DESCRIPTION>
    <LASTDATE>datetime</LASTDATE>
    <USERID>string</USERID>
    <PHASETYPE>string</PHASETYPE>
    <SQFT>numeric</SQFT>
    <STRTDATE>datetime</STRTDATE>
    <CMPLDATE>datetime</CMPLDATE>
    <ACTVDATE>datetime</ACTVDATE>
    <JC_COSTLIST>string</JC_COSTLIST>
    …{additional custom field tags}
  </JC_PHASE>
  …
<IRESMRIJCPhaseReadResponse>
```

errorInfo (error response)

```
<IRESMRIJCPhaseReadErrorMessage>
  <error>
    <message>
      <IRESMRIJCPhaseReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIJCPhaseReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIJCPhaseReadErrorMessage>
```

Phase Update Interface (IRESMRIJCPhaseUpdate)

This method provides the ability to update the job cost phase information in the JC_PHASE table in an MRI database. This method is available for all installations that include a license for the JobCost application. If a record already exists for a provided JOBCODE/JC_PHASECODE, the information in the record will be updated with that in the request. If a record does not already exist for a provided JOBCODE/JC_PHASECODE, a new record will be inserted into the table. The request will **not** be treated as a batch. As a result, all valid rows will be processed, and the database will be updated with errors returned for unsuccessful items. This method will provide support for additional custom fields, foreign key validation, and defaulting based on the definition in the MRIField table. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

```
<IRESMRIJCPhaseUpdate>
  <JC_PHASE>
    <JOBCODE>string</JOBCODE>                    Required
    <JC_PHASECODE>string</JC_PHASECODE>          Required
    <DESCRIPTION>string</DESCRIPTION>            Req'd for Insert
    <LASTDATE>datetime</LASTDATE>                Blocked
    <USERID>string</USERID>                      Blocked
    <PHASETYPE>string</PHASETYPE>                Optional
    <SQFT>numeric</SQFT>                         Optional
    <STRTDATE>datetime</STRTDATE>                Optional
    <CMPLDATE>datetime</CMPLDATE>                Optional
    <ACTVDATE>datetime</ACTVDATE>                Optional
    <JC_COSTLIST>string</JC_COSTLIST>            Optional
…{additional custom field tags}                  Optional
  </JC_PHASE>
</IRESMRIJCPhaseUpdate>
```

Note

Additional custom field tags are mapped to custom fields in the JC_PHASE table by the field name. No additional validation will be performed beyond the foreign key validation with default values being set from the MRIFIELD definition where provided.

returnedInfo (API response)

```
<IRESMRIJCPhaseResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsUpdated>numeric</rowsUpdated>
  <rowsInserted>numeric</rowsInserted>
  <rowsValidated>numeric</rowsValidated>
  <rowsInError>numeric</rowsInError>
</IRESMRIJCPhaseResponse>
```

errorInfo (error response)

```
<IRESMRIJCPhaseUpdateErrorMessage>
  <error>
    <message>
      <IRESMRIJCPhaseUpdateError>
        <error>
          <JOBCODE>string</JOBCODE>              Where available
          <JC_PHASECODE>string</JC_PHASECODE>    Where available
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIJCPhaseUpdateError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIJCPhaseUpdateErrorMessage>
```

Cost Code List Read Interface (IRESMRIJCCostListRead)

This method provides the ability to read cost code list information from the Cost Lists table (JC_COSTLIST) in an MRI database. This method is available for all installations that include a license for the JobCost application. All fields from the JC_COSTLIST table, plus additional custom fields, will be returned in the response. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRIJCCostListRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRIJCCostListRead>
```

Overlay 2—A single row for the specified cost list code.

```
<IRESMRIJCCostListRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <JC_COSTLIST>string</JC_COSTLIST>             Required
</IRESMRIJCCostListRead>
```

Overlay 3—All rows updated since a specified last update date.

```
<IRESMRIJCCostListRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LASTDATE>datetime</LASTDATE>                 Required
</IRESMRIJCCostListRead>
```

returnedInfo (API response)

```
<IRESMRIJCCostListReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>999</rowsReturned>
  <JC_COSTLIST>
    <JC_COSTLIST>string</JC_COSTLIST>
    <DESCRIPTION>string</DESCRIPTION>
    <LASTDATE>datetime</LASTDATE>
    <USERID>string</USERID>
    <MAJORLGT>numeric</MAJORLGT>
    <CODEDELIM>string</CODEDELIM>
    <SUPPID1>string</SUPPID1>
    <SUPPID2>string</SUPPID2>
    …{additional custom field tags}
  </JC_COSTLIST>
  …
<IRESMRIJCCostListReadResponse>
```

errorInfo (error response)

```
<IRESMRIJCCostListReadErrorMessage>
  <error>
    <message>
      <IRESMRIJCCostListReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIJCCostListReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIJCCostListReadErrorMessage>
```

Cost Code Read Interface (IRESMRIJCCostCodeRead)

This method provides the ability to read job cost list information from the Job Cost Codes table (JC_COSTCD) in an MRI database. This method is available for all installations that include a license for the JobCost application. All fields from the JC_COSTCD table, plus additional custom fields, will be returned in the response. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRIJCCostCodeRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRIJCCostCodeRead>
```

Overlay 2—All rows for the specified cost list.

```
<IRESMRIJCCostCodeRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <JC_COSTLIST>string</JC_COSTLIST>             Required
</IRESMRIJCCostCodeRead>
```

Overlay 3—A single row for the specified cost list and cost code.

```
<IRESMRIJCCostCodeRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <JC_COSTLIST>string</JC_COSTLIST>             Required
  <JC_COSTCODE>string</JC_COSTCODE>             Required
</IRESMRIJCCostCodeRead>
```

Overlay 4—All rows updated since a specified last update date.

```
<IRESMRIJCCostCodeRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LASTDATE>datetime</LASTDATE>                 Required
</IRESMRIJCCostCodeRead>
```

returnedInfo (API response)

```
<IRESMRIJCCostCodeReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>999</rowsReturned>
  <JC_COSTCD>
    <JC_COSTLIST>string</JC_COSTLIST>
    <JC_COSTCODE>string</JC_COSTCODE>
    <CODENAME>string</CODENAME>
    <LASTDATE>datetime</LASTDATE>
    <USERID>string</USERID>
    <ROLLUP>string</ROLLUP>
    <CATEGORY>string</CATEGORY>
    <MAJOR>string</MAJOR>
    <MINOR>string</MINOR>
    <UNITS>string</UNITS>
    <COSTTYPE>string</COSTTYPE>
    <SUPPID1>string</SUPPID1>
    <SUPPCODE1>string</SUPPCODE1>
    <SUPPID2>string</SUPPID2>
    <SUPPCODE2>string</SUPPCODE2>
    <LEDGCODE>string</LEDGCODE>
    <ACCTNUM>string</ACCTNUM>
```

```
    …{additional custom field tags}
  </JC_COSTCD>
  …
<IRESMRIJCCostCodeReadResponse>
```

errorInfo (error response)

```
<IRESMRIJCCostCodeReadErrorMessage>
  <error>
    <message>
      <IRESMRIJCCostCodeReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIJCCostCodeReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIJCCostCodeReadErrorMessage>
```

Cost Category Read Interface (IRESMRIJCCostCategoryRead)

This method provides the ability to read cost category information from the Job Cost Categories table (JC_COSTCAT) in an MRI database. This method is available for all installations that include a license for the JobCost application. All fields from the JC_COSTCAT table, plus additional custom fields, will be returned in the response. The capitalized field name will be used as the tag name for all fields unless otherwise indicated. The parameter sets below define how standard fields will be processed.

commandInfo (request)

Overlay 1—All rows.

```
<IRESMRIJCCostCategoryRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <all-rows>Y</all-rows>                        Required
</IRESMRIJCCostCategoryRead>
```

Overlay 2—All rows for the specified cost list and cost code.

```
<IRESMRIJCCostCategoryRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <JC_COSTLIST>string</JC_COSTLIST>             Required
  <JC_COSTCODE>string</JC_COSTCODE>             Required
</IRESMRIJCCostCategoryRead>
```

Overlay 3—A single row for the specified cost list, cost code, and cost category.

```
<IRESMRIJCCostCategoryRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <JC_COSTLIST>string</JC_COSTLIST>             Required
  <JC_COSTCODE>string</JC_COSTCODE>             Required
  <CATEGORY>string</CATEGORY>                   Required
</IRESMRIJCCostCategoryRead>
```

Overlay 4—All rows updated since a specified last update date.

```
<IRESMRIJCCostCategoryRead>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <LASTDATE>datetime</LASTDATE>                 Required
</IRESMRIJCCostCategoryRead>
```

returnedInfo (API response)

```
<IRESMRIJCCostCategoryReadResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsReturned>999</rowsReturned>
  <JC_COSTCAT>
    <JC_COSTLIST>string</JC_COSTLIST>
    <JC_COSTCODE>string</JC_COSTCODE>
    <CATEGORY>string</CATEGORY>
    <DESCRIPTION>string</DESCRIPTION>
    <LASTDATE>datetime</LASTDATE>
    <USERID>string</USERID>
    …{additional custom field tags}
  </JC_COSTCAT>
  …
<IRESMRIJCCostCategoryReadResponse>
```

errorInfo (error response)

```
<IRESMRIJCCostCategoryReadErrorMessage>
  <error>
    <message>
      <IRESMRIJCCostCategoryReadError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIJCCostCategoryReadError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</IRESMRIJCCostCategoryReadErrorMessage>
```

Residential Management APIs

Guest Card Creation (MRIRMGuestCardCreate)

This method provides the ability to create guest card entries in an MRI database. This method is available for all installations with a valid license for the Residential Management application. No support is provided through this method for updating existing guest cards. The request will **not** be treated as a batch. As a result, all valid rows will be processed, and the database will be updated with errors returned for unsuccessful items. The parameter sets below define how standard fields will be processed. The custom field tag can be used for any nodes found in the following tables: Resident Names (NAME), !{RM Resident} Addresses (RMADDR), !{RM Resident} Actions (RMACTION), and Prospect-Applicant Information (PROSPECT).

commandInfo (request)

```
<MRIRMGuestCardCreate>
  <interfaceVersion>1.0</interfaceVersion>      Optional
  <SAVEGUESTCARDINFO>
    <NAMEID>string</NAMEID>                     Optional
    <RMPROPID>string</RMPROPID>                 Optional
    <FIRSTNAME>string</FIRSTNAME>               Optional
    <LASTNAME>string</LASTNAME>                 Required
    <EMAIL>string</EMAIL>                       Optional
    <PHONEDAY>string</PHONEDAY>                 Optional
    <COMMENT>string</COMMENT>                   Optional
    <MRKTID>string</MRKTID>                     Optional
    <ADDRESS1>string</ADDRESS1>                 Optional
    <ADDRESS2>string</ADDRESS2>                 Optional
    <CITY>string</CITY>                         Optional
    <STATE>string</STATE>                       Optional
    <ZIP>string</ZIP>                           Optional
    <PHONEEVENING>string</PHONEEVENING>         Optional
    <ESTMOVEIN>DateTime</ESTMOVEIN>             Optional
    <USERID>string</USERID>                     Optional
    <RECORDTYPE>string</RECORDTYPE>             Optional
    <TRDATE>DateTime</TRDATE>                   Optional
    <CUSTOMFIELD TABLE=”TABLENAME” FIELD=”CUSTOMNAME”>
     string
    </CUSTOMFIELD>                              Optional
  </SAVEGUESTCARDINFO>
  …
</MRIRMGuestCardCreate>
```

returnedInfo (API response)

```
<MRIRMGuestCardCreateResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsInserted>numeric</rowsInserted>
  <rowsInError>numeric</rowsInError>
</MRIRMGuestCardCreateResponse>
```

errorInfo (error response)

```
<MRIRMGuestCardCreateErrorMessage>
  <error>
    <message>
      <MRIRMGuestCardCreateError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </MRIRMGuestCardCreateError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
  …
</MRIRMGuestCardCreateErrorMessage>
```

Resident Interface (IRESMRIRMResidentEdit)

This method provides the ability for an alternate property management service to add resident information sufficient for a client to produce a Rent Roll report (MRI_RMROLL) and a Unit Status report (MRI_UNITSTAT) and view current recurring charges. This method is available for all installations with a valid license for the Residential Management application. The request will not be treated as a batch. As a result, all valid rows will be processed, and the database will be updated with errors returned for unsuccessful items. The parameter sets below define how standard fields will be processed.

commandInfo (request)

```
<IRESMRIRMResidentEdit>
  <interfaceVersion>1.0</interfaceVersion>            Optional
  <ACTION>string</ACTION>                             Required
  <APICALLER>string</APICALLER>                       Optional
  <RMPROPID>string</RMPROPID>
  <RMBLDGID>string</RMBLDGID>                         Required
  <UNITID>string</UNITID>                             Required
  <OCCUPANTS>
    <OCCUPANT>
      <NAMEID>string</NAMEID>                         TBD
      <NAMEGROUP>string</NAMEGROUP>                   TBD
      <LASTNAME>string</LASTNAME>                     Required
```

```
      <FIRSTNAME>string</FIRSTNAME>                          Required
      <MIDINIT>string</MIDINIT>                              Optional
      <STATUS>HEAD</STATUS>                                  Required
      <BIRTHDAY>date</BIRTHDAY>                              Optional
      <SSNO>string</SSNO>                                    Optional
      <LICENSE>string</LICENSE>                              Optional
      <PHONE>string</PHONE>                                  Optional
      <COMMENT>string</COMMENT>                              Optional
      <SEX>string</SEX>                                      Optional
      <OCCUPATION>string</OCCUPATION>                        Optional
      <INCOME>string</INCOME>                                Optional
      <LOCATION>string</LOCATION>                            Optional
      <MRKTID>string</MRKTID>                                Optional
      <LEASEAGID>string</LEASEAGID>                          Optional
      <COMPANY>string</COMPANY>                              Optional
      <JOBTITLE>string</JOBTITLE>                            Optional
      <NONSF>string</NONSF>                                  Optional
      <NOLATE>string</NOLATE>                                Optional
      <PAYALLOWED>string</PAYALLOWED>                        Optional
      <MARRIED>string</MARRIED>                              Optional
      <EMAIL>string</EMAIL>                                  Optional
      <CREDITSTATUS>string</CREDITSTATUS>                    Optional
      <PASSTID>string</PASSTID>                              Optional
      <OCCUPYDATE>date</OCCUPYDATE>                          Required
      <VACATEDATE>date</VACATEDATE>                          Optional
      <WORKPHONE1>string</WORKPHONE1>                        Optional
      <WORKPHONE2>string</WORKPHONE2>                        Optional
      <CELL>string</CELL>                                    Optional
      <FAX>string</FAX>                                      Optional
      <MVIREASN>string</MVIREASN>                            Optional
      <NMBRRESIDENTS>number</NMBRRESIDENTS>                  Optional
      <NOSSNO>string</NOSSNO>                                Optional
      <BACKGRNDSTATUS>string</BACKGRNDSTATUS>                Optional
      <OTRESBCKGRNDREQ>string</OTRESBCKGRNDREQ>              Optional
      <WEBNOCHECKS>string</WEBNOCHECKS>                      Optional
      <OCCINACTIVE>string</OCCINACTIVE>                      Optional
      <AUTOSPREAD>string</AUTOSPREAD>                        Optional
      <CHKOPTOUT>string</CHKOPTOUT>                          Optional
      <EXCLUDENWP>string</EXCLUDENWP>                        Optional
      <NWPDATE />
    </OCCUPANT>
  </OCCUPANTS>
  …
</IRESMRIRMResidentEdit>
```

returnedInfo (API response)

```
<IRESMRIRMResidentEditResponse>
  <interfaceVersion>1.0</interfaceVersion>
  <rowsInserted>numeric</rowsInserted>
  <rowsInError>numeric</rowsInError>
</IRESMRIRMResidentEditResponse>
```

errorInfo (error response)

```
<IRESMRIRMResidentEditErrorMessage>
  <error>
    <message>
      <IRESMRIRMResidentEditError>
        <error>
          <message>string</message>
          <errorNumber>numeric</errorNumber>
        </error>
      </IRESMRIRMResidentEditError>
    </message>
    <errorNumber>numeric</errorNumber>
  </error>
</IRESMRIRMResidentEditErrorMessage>
```

Solution Integrations 5

This chapter is intended for solution providers who are building integrations with MRI as part of the Partner Connect program.

Utility Billing Integration

MRI's utility billing integration uses an API to allow property, unit, lease, and resident data to be retrieved from a client's database. This data will be used by vendors to calculate what utility charge amounts need to be billed for a property's residents. The MRI utility APIs can be used with any utility billing system, such as Ratio Utility Billing System (RUBS). RUBS is a method of calculating a resident's utility bill based on a combination of factors, which can include occupancy, apartment square footage, and number of beds. This method is an alternate solution to having a property sub metered.

MRI has updated the utility APIs to be MITS 3.0 compliant, so all data being transferred is standard and applicable to all partners. With the 3.0 version, users will have access to all relevant messages and errors that occur during a data transfer, located under **Setup and Maintenance > Administrative Options > API Message Inquiry**. Additionally, a reconciliation report called the Resident Billing API Exception report (WEB_RUBSREC) has been created that can show users how many utility charges posted during a given date range versus which charges failed to post during a data transfer. This report can also be used to historically track utility billing charges for research purposes.

MRI supports MITS 3.0 and 2.0 to maintain maximum flexibility and backwards compatibility. To download MITS 3.0 Resident Transactions schema, go to the following URL:

http://www.mitsproject.com/Content/ServeContent.cfm?menuID=463&ContentItemID=2890

Note

Because MITS standards are not specific to utility billing, not all parts of the MITS schema are followed.

Before you call RUBS APIs, configure the following settings in MRI for Windows under **Management Options > General**:

- **Residential Utility Billing Service**—This check box must be selected in order to call RUBS APIs.
- **RUBS Submission XML Directory**—This field is not used with the RUBS APIs.
- **Automatically Post Transactions**—Select this check box to automatically post all utility batches sent through the UtilityCreate API.

- **Use Virtual Site**—Select this check box to look at the DP Activity table (DPACTIVITY) and see what sites are tied to the property that utility charges are created for, and then append the site to the RMBATCHID.
- **Retain Service Dates**—Select this check box to create service date records in the Utility Billing Service Dates table (TB_RM_SERVICEDATES). Service dates are inputted in the Create API.

MITS 3.0

The UtilityRead and UtilityCreate APIs will not process requests as batches. When multiple properties are included in one Read request, if one errors out, the rest will still return any valid results. For the Create APIs, all transactions will be posted until a batch processing error occurs. Posted transactions will not revert when a batch processing error is received, but you will need to resend any transactions that were intended to be processed after the error. Errors other than a batch processing error will not stop the create process for valid charges.

MITS 3.0 UtilityRead

The UtilityRead API reads property, unit, lease, resident, and utility charge data from an MRI client database that will be used to perform RUBS calculations.

```
<MRIRMUtilityReadMITS30>
  <ResidentTransactions>                                    Required
    <Property>                                              Required
      <PropertyID>                                          Required
        <Identification IDType="RMPROPID">                  Required
          <IDValue>string</IDValue>                         Required
        </Identification>
        <MarketingName>string</MarketingName>               Required
      </PropertyID>
    </Property>
  </ResidentTransactions>
</MRIRMUtilityReadMITS30>
```

MITS 3.0 UtilityRead Response

```
<ResidentTransactions>
  <Property>
    <PropertyID>
      <Identification IDType="RMPROPID">
        <IDValue>string</IDValue>
      </Identification>
      <MarketingName>string</MarketingName>
      <Address AddressType="property">
        <Address>string</Address>
        <City>string</City>
        <State>string</State>
        <PostalCode>string</PostalCode>
```

```
  </Address>
</PropertyID>
<RT_Customer>
  <Identification IDType="NAMEGROUP">
    <IDValue>string</IDValue>
  </Identification>
  <Customers>
    <Customer>
      <Identification IDType="NAMEID">
        <IDValue>string</IDValue>
      </Identification>
      <Name>
        <FirstName>string</FirstName>
        <LastName>string</LastName>
      </Name>
      <Address AddressType="billing">
        <Address>string</Address>
        <City>string</City>
        <State>string</State>
        <PostalCode>string</PostalCode>
        <Email>string</Email>
      </Address>
      <Phone PhoneType="home">
        <PhoneNumber>string</PhoneNumber>
      </Phone>
      <Lease>
        <Identification IDType="RMLEASE">
          <IDValue>string</IDValue>
        </Identification>
        <ExpectedMoveOutDate>YYYY-MM-DD</ExpectedMoveOutDate>
        <LeaseFromDate>YYYY-MM-DD</LeaseFromDate>
        <LeaseToDate>YYYY-MM-DD</LeaseToDate>
        <ActualMoveIn>YYYY-MM-DD</ActualMoveIn>
        <ResponsibleForLease>
         ”true” or “false”
        </ResponsibleForLease>
      </Lease>
    </Customer>
  </Customers>
  <Unit>
    <Identification IDType="UNITID">
      <IDValue>string</IDValue>
    </Identification>
    <Identification IDType="RMBLDGID">
      <IDValue>string</IDValue>
    </Identification>
    <Identification IDType="RMPROPID">
      <IDValue>string</IDValue>
    </Identification>
    <UnitType>string</UnitType>
    <UnitBedrooms>decimal</UnitBedrooms>
    <UnitBathrooms>decimal</UnitBathrooms>
    <MinSquareFeet>integer</MinSquareFeet>
    <MaxSquareFeet>integer</MaxSquareFeet>
    <UnitRent>decimal</UnitRent>
    <NumberOccupants Total="2" Child="0">
     complex
```

```
        </NumberOccupants>
        <FloorplanName>string</FloorplanName>
        <Address AddressType="property">
          <Address>string</Address>
          <City>string</City>
          <State>string</State>
          <PostalCode>string</PostalCode>
        </Address>
      </Unit>
      <RTServiceTransactions>
        <Services ServicesType="string">
          <Detail>
            <Service ServiceType="string" />
            <ChargeCode>string</ChargeCode>
          </Detail>
        </Services>
      </RTServiceTransactions>
    </RT_Customer>
  </Property>
</ResidentTransactions>
```

Note

If an issue exists with how the data is set up at a certain level, parts of this response will not return certain tags. For example, if no utility charges are mapped for this property, then the service transactions XML will be empty.

You will see the following response if the UtilityRead call fails because the schema is not compliant or RUBS has not been properly configured:

```
<ResidentTransactions>
  <Property>
    <PropertyID>
      <Identification IDType="Status">
        <IDValue>Fail</IDValue>
      </Identification>
      <MarketingName />
    </PropertyID>
  </Property>
</ResidentTransactions>
```

MITS 3.0 UtilityRead Error Response

Note

Some error response records can have both successful responses and error responses. You should review the errors you receive to determine whether it is a critical error or a warning.

```
<MRIRMUtilityReadMITS30ErrorMessage>
  <error>
    <message>string</message>
    <errorNumber>numeric</errorNumber>
  </error>
</MRIRMUtilityReadMITS30ErrorMessage>
```

MITS 3.0 UtilityCreate

With the UtilityCreate API, vendors can send utility charges to an MRI client database after the calculations are finished. This API can only be called asynchronously. For more information on calling APIs asynchronously, refer to “Calling APIs Asynchronously” on page 14.

```
<MRIRMUtilityCreateMITS30>                                  Optional
  <ResidentTransactions>                                    Required
    <Property>                                              Required
      <PropertyID>                                          Required
        <Identification IDType="RMPROPID">                  Required
          <IDValue>string</IDValue>                         Required
        </Identification>
        <MarketingName>string</MarketingName>               Required
        <Address AddressType="property">
          <Address>string</Address>
          <City>string</City>
          <State>string</State>
          <PostalCode>string</PostalCode>
        </Address>
      </PropertyID>
      <RT_Customer>                                         Required
        <Identification IDType="NAMEGROUP">                 Required
          <IDValue>string</IDValue>                         Required
        </Identification>
        <Customers>
          <Customer>
            <Identification IDType="NAMEID">
              <IDValue>string</IDValue>
            </Identification>
            <Name>
              <FirstName>string</FirstName>
              <LastName>string</LastName>
            </Name>
            <Address AddressType="billing">
              <Address>string</Address>
              <City>string</City>
              <State>string</State>
              <PostalCode>string</PostalCode>
              <Email>string</Email>
            </Address>
            <Phone PhoneType="home">
              <PhoneNumber>numeric</PhoneNumber>
            </Phone>
            <Lease>
              <Identification IDType="RMLEASE">
                <IDValue>string</IDValue>
              </Identification>
              <ExpectedMoveOutDate>
               YYYY-MM-DD
              </ExpectedMoveOutDate>
              <LeaseFromDate>YYYY-MM-DD</LeaseFromDate>
              <LeaseToDate>YYYY-MM-DD</LeaseToDate>
              <ActualMoveIn>YYYY-MM-DD</ActualMoveIn>
              <ResponsibleForLease>”true” or “false”
```

```
            </ResponsibleForLease>
          </Lease>
        </Customer>
      </Customers>
      <Unit>                                          Required
        <Identification IDType="UNITID">              Required
          <IDValue>string</IDValue>                   Required
        </Identification>
        <Identification IDType="RMBLDGID">            Required
          <IDValue>string</IDValue>                   Required
        </Identification>
        <Identification IDType="RMPROPID">            Required
          <IDValue>string</IDValue>                   Required
        </Identification>
        <UnitType>string</UnitType>
        <UnitBedrooms>decimal</UnitBedrooms>
        <UnitBathrooms>decimal</UnitBathrooms>
        <MinSquareFeet>integer</MinSquareFeet>
        <MaxSquareFeet>integer</MaxSquareFeet>
        <UnitRent>decimal</UnitRent>
        <NumberOccupants Total="2" Child="0">complex
        </NumberOccupants>
        <FloorplanName>string</FloorplanName>
        <Address AddressType="property">
          <Address>string </Address>
          <City>string</City>
          <State>string</State>
          <PostalCode>string</PostalCode>
        </Address>
      </Unit>
      <RTServiceTransactions>                         Required
        <Services ServicesType="string">
          <Detail>
            <Service ServiceType="string" />
            <ChargeCode>string</ChargeCode>
          </Detail>
        </Services>
        <Transactions>                                Required
          <Charge>                                    Required
            <Detail>                                  Required
              <ServiceToDate>YYYY-MM-DD
              </ServiceToDate>                        Required
              <ServiceFromDate>YYYY-MM-DD
              </ServiceFromDate>                      Required
              <ChargeCode>string</ChargeCode>         Required
              <Amount>decimal</Amount>                Required
            </Detail>
          </Charge>
        </Transactions>
      </RTServiceTransactions>
    </RT_Customer>
  </Property>
 </ResidentTransactions>
</MRIRMUtilityCreateMITS30>
```

MITS 3.0 UtilityCreate Response

You will receive the following response after the API has finished processing asynchronously:

Note

For more information on how to retrieve responses, refer to “Calling APIs Asynchronously” on page 14.

```
<?xml version="1.0" encoding="utf-8"?>
<MRIRMUtilityCreateMITS30Response Status="Pass">
  <TransactionProcessingResults>
    <TotalNumberOfPropertiesInRequest>
     numeric
    </TotalNumberOfPropertiesInRequest>
    <NumberOfPropertiesProcessed>
     numeric
    </NumberOfPropertiesProcessed>
    <NumberOfPropertiesNotProcessed>
     numeric
    </NumberOfPropertiesNotProcessed>
    <SummaryMessages />
    <PropertySummaries>
      <PropertySummary>
        <PropertyID>string</PropertyID>
        <TotalNumberOfResidentsInRequest>
         numeric
        </TotalNumberOfResidentsInRequest>
        <NumberOfResidentsProcessedWithOutError>
         numeric
        </NumberOfResidentsProcessedWithOutError>
        <NumberOfResidentsProcessedWithError>
         numeric
        </NumberOfResidentsProcessedWithError>
        <NumberOfTransactionsInRequest>
         numeric
        </NumberOfTransactionsInRequest>
        <NumberOfTransactionsProcessedWithOutError>
         numeric
        </NumberOfTransactionsProcessedWithOutError>
        <NumberOfTransactionsProcessedWithError>
         numeric
        </NumberOfTransactionsProcessedWithError>
        <BatchId>string</BatchId>
        <TotalAmountOfChargesPosted xsi:type="xsd:decimal">
         numeric
        </TotalAmountOfChargesPosted>
        <TransactionMessages />
      </PropertySummary>
    </PropertySummaries>
  </TransactionProcessingResults>
</MRIRMUtilityCreateMITS30Response>
```

MITS 3.0 UtilityCreate Error Response

```
<MRIRMUtilityCreateMITS30ErrorMessage>
  <error>
    <message>string</message>
    <errorNumber>numeric</errorNumber>
  </error>
</MRIRMUtilityCreateMITS30ErrorMessage>
```

MITS 3.0 UtilityCreate Property Level Error Response

```
<?xml version="1.0" encoding="utf-8"?>
<MRIRMUtilityCreateMITS30Response Status="Fail">
  <TransactionProcessingResults>
    <TotalNumberOfPropertiesInRequest>
     numeric
    </TotalNumberOfPropertiesInRequest>
    <NumberOfPropertiesProcessed>
     numeric
    </NumberOfPropertiesProcessed>
    <NumberOfPropertiesNotProcessed>
     numeric
    </NumberOfPropertiesNotProcessed>
    <SummaryMessages>
      <Message>string</Message>
    </SummaryMessages>
    <PropertySummaries />
  </TransactionProcessingResults>
</MRIRMUtilityCreateMITS30Response>
```

MITS 3.0 UtilityCreate Transaction Level Error Response

```
<?xml version="1.0" encoding="utf-8"?>
<MRIRMUtilityCreateMITS30Response Status="Fail">
  <TransactionProcessingResults>
    <TotalNumberOfPropertiesInRequest>
     numeric
    </TotalNumberOfPropertiesInRequest>
    <NumberOfPropertiesProcessed>
     numeric
    </NumberOfPropertiesProcessed>
    <NumberOfPropertiesNotProcessed>
     numeric
    </NumberOfPropertiesNotProcessed>
    <SummaryMessages/>
    <PropertySummaries>
      <PropertySummary>
        <PropertyID>numeric</PropertyID>
        <TotalNumberOfResidentsInRequest>
         numeric
        </TotalNumberOfResidentsInRequest>
        <NumberOfResidentsProcessedWithOutError>
         numeric
```

```
          </NumberOfResidentsProcessedWithOutError>
          <NumberOfResidentsProcessedWithError>
           numeric
          </NumberOfResidentsProcessedWithError>
          <NumberOfTransactionsInRequest>
           numeric
          </NumberOfTransactionsInRequest>
          <NumberOfTransactionsProcessedWithOutError>
           numeric
          </NumberOfTransactionsProcessedWithOutError>
          <NumberOfTransactionsProcessedWithError>
           numeric
          </NumberOfTransactionsProcessedWithError>
          <BatchId>string</BatchId>
          <TransactionMessages>
            <TransactionMessage>
              <PropertyId>string</PropertyId>
              <BuildingId>string</BuildingId>
              <UnitId>string</UnitId>
              <NameId>string</NameId>
              <ChargeCode>string</ChargeCode>
              <PassedValidation>false</PassedValidation>
              <Message>string</Message>
              <Tranid/>
            </TransactionMessage>
          </TransactionMessages>
        </PropertySummary>
      </PropertySummaries>
    </TransactionProcessingResults>
</MRIRMUtilityCreateMITS30Response>
```

MITS 2.0

To download MITS 2.0 Resident Transactions schema, go to the following URL:

http://www.mitsproject.com/Content/ServeContent.cfm?menuID=463&ContentItemID=2915

MITS 2.0 UtilityRead

```
<MRIRMUtilityRead>                                          Required
  <PrimaryID>RMA244</PrimaryID>                             Required
</MRIRMUtilityRead>
```

MITS 2.0 UtilityRead Response

```
<MRIRMUtilityReadResponse>
  <ResidentTransactions
   xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
   xmlns:MITS=http://my-company.com/namespace
   xmlns="CIS.Integration.Multifamily.Schema.MITS.MITS
ResidentTransactions2_0">
```

```
<Property>
  <PropertyID>
    <MITS:Identification Type="apt">
      <MITS:PrimaryID>string</MITS:PrimaryID>
      <MITS:MarketingName>string</MITS:MarketingName>
    </MITS:Identification>
    <MITS:Address Type="property">
      <MITS:Address1>string</MITS:Address1>
      <MITS:Address2>string</MITS:Address2>
      <MITS:City>string</MITS:City>
      <MITS:State> string </MITS:State>
      <MITS:PostalCode> string </MITS:PostalCode>
    </MITS:Address>
  </PropertyID>
  <RT_Customer>
    <CustomerID> string </CustomerID>
    <Customers>
      <MITS:Customer Type="current">
        <MITS:CustomerID> string </MITS:CustomerID>
        <MITS:Property>
          <MITS:PrimaryID> string </MITS:PrimaryID>
          <MITS:MarketingName> string </MITS:MarketingName>
          <MITS:UnitID> string </MITS:UnitID>
          <MITS:BuildingID> string </MITS:BuildingID>
        </MITS:Property>
        <MITS:Name>
          <MITS:FirstName> string </MITS:FirstName>
          <MITS:LastName> string </MITS:LastName>
        </MITS:Name>
        <MITS:Address Type="billing">
          <MITS:Address1> string </MITS:Address1>
          <MITS:City> string </MITS:City>
          <MITS:State> string </MITS:State>
          <MITS:PostalCode> string </MITS:PostalCode>
          <MITS:Email> string </MITS:Email>
        </MITS:Address>
        <MITS:Phone Type="home">
          <MITS:PhoneNumber>string</MITS:PhoneNumber>
        </MITS:Phone>
        <MITS:Lease>
          <MITS:ExpectedMoveOutDate>
           MM/DD/YYYY
          </MITS:ExpectedMoveOutDate>
          <MITS:LeaseFromDate>
           MM/DD/YYYY
          </MITS:LeaseFromDate>
          <MITS:LeaseToDate>MM/DD/YYYY</MITS:LeaseToDate>
          <MITS:ActualMoveIn>MM/DD/YYYY</MITS:ActualMoveIn>
          <MITS:ActualMoveOut>MM/DD/YYYY</MITS:ActualMoveOut>
          <MITS:ResponsibleforLease>
           MM/DD/YYYY
          </MITS:ResponsibleforLease>
        </MITS:Lease>
      </MITS:Customer>
    </Customers>
    <RT_Unit>
      <UnitID> string</UnitID>
```

```
            <NumberOccupants Total="1" />
            <Unit>
              <MITS:Information>
                <MITS:UnitID>string</MITS:UnitID>
                <MITS:UnitType>string</MITS:UnitType>
                <MITS:UnitBedrooms>decimal</MITS:UnitBedrooms>
                <MITS:MinSquareFeet>integer</MITS:MinSquareFeet>
                <MITS:MaxSquareFeet>integer</MITS:MaxSquareFeet>
                <MITS:UnitRent>decimal(7-2 decimal)</MITS:UnitRent>
                <MITS:Address Type="property">
                  <MITS:Address1>string</MITS:Address1>
                  <MITS:City>string</MITS:City>
                  <MITS:State>string</MITS:State>
                  <MITS:PostalCode>string</MITS:PostalCode>
                </MITS:Address>
              </MITS:Information>
            </Unit>
          </RT_Unit>
          <RTServiceTransactions>
            <Services Type="Telephone">
              <Detail>
                <Service Type="Telephone" />
                <Chargecode>TEL</Chargecode>
              </Detail>
            </Services>
          </RTServiceTransactions>
        </RT_Customer>
      </Property>
    </ResidentTransactions>
</MRIRMUtilityReadResponse>
```

MITS 2.0 UtilityCreate

```
<MRIRMUtilityCreate>
  <ResidentTransactions
   xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
   xmlns:MITS=http://my-company.com/namespace
   xmlns="CIS.Integration.Multifamily.Schema.MITS.MITS
ResidentTransactions2_0">
    <Property>
      <PropertyID>
        <MITS:Identification Type="apt">
          <MITS:PrimaryID>string</MITS:PrimaryID>
          <MITS:MarketingName>string</MITS:MarketingName>
        </MITS:Identification>
        <MITS:Address Type="property">
          <MITS:Address1>string</MITS:Address1>
          <MITS:Address2>string</MITS:Address2>
          <MITS:City>string</MITS:City>
          <MITS:State> string </MITS:State>
          <MITS:PostalCode> string </MITS:PostalCode>
        </MITS:Address>
      </PropertyID>
      <RT_Customer>
        <CustomerID> string </CustomerID>
```

```
<Customers>
  <MITS:Customer Type="current">
    <MITS:CustomerID> string </MITS:CustomerID>
    <MITS:Property>
      <MITS:PrimaryID> string </MITS:PrimaryID>
      <MITS:MarketingName> string </MITS:MarketingName>
      <MITS:UnitID> string </MITS:UnitID>
      <MITS:BuildingID> string </MITS:BuildingID>
    </MITS:Property>
    <MITS:Name>
      <MITS:FirstName> string </MITS:FirstName>
      <MITS:LastName> string </MITS:LastName>
    </MITS:Name>
    <MITS:Address Type="billing">
      <MITS:Address1> string </MITS:Address1>
      <MITS:City> string </MITS:City>
      <MITS:State> string </MITS:State>
      <MITS:PostalCode> string </MITS:PostalCode>
      <MITS:Email> string </MITS:Email>
    </MITS:Address>
    <MITS:Phone Type="home">
      <MITS:PhoneNumber> string </MITS:PhoneNumber>
    </MITS:Phone>
    <MITS:Lease>
      <MITS:ExpectedMoveOutDate>
       MM/DD/YYYY
      </MITS:ExpectedMoveOutDate>
      <MITS:LeaseFromDate>MM/DD/YYYY</MITS:LeaseFromDate>
      <MITS:LeaseToDate>MM/DD/YYYY</MITS:LeaseToDate>
      <MITS:ActualMoveIn>MM/DD/YYYY</MITS:ActualMoveIn>
      <MITS:ActualMoveOut>MM/DD/YYYY</MITS:ActualMoveOut>
      <MITS:ResponsibleforLease>
       MM/DD/YYYY
      </MITS:ResponsibleforLease>
    </MITS:Lease>
  </MITS:Customer>
</Customers>
<RT_Unit>
  <UnitID>string</UnitID>
  <NumberOccupants Total="2" />
  <Unit>
    <MITS:Information>
      <MITS:UnitID> string </MITS:UnitID>
      <MITS:UnitType> string </MITS:UnitType>
      <MITS:UnitBedrooms>decimal</MITS:UnitBedrooms>
      <MITS:MinSquareFeet>integer</MITS:MinSquareFeet>
      <MITS:MaxSquareFeet>integer</MITS:MaxSquareFeet>
      <MITS:UnitRent>
       decimal (7-2 decimal)
      </MITS:UnitRent>
      <MITS:Address Type="property">
        <MITS:Address1>string</MITS:Address1>
        <MITS:City>string</MITS:City>
        <MITS:State>string</MITS:State>
        <MITS:PostalCode>string</MITS:PostalCode>
      </MITS:Address>
    </MITS:Information>
```

```
          </Unit>
        </RT_Unit>
        <RTServiceTransactions>
          <Services Type="Water">
            <Detail>
              <Service Type="Water" />
              <Chargecode>string</Chargecode>
            </Detail>
          </Services>
          <Transactions>
            <Charge>
              <Detail>
                <ServiceToDate>MM/DD/YYYY</ServiceToDate>
                <ServiceFromDate>MM/DD/YYYY</ServiceFromDate>
                <ChargeCode>string</ChargeCode>
                <Amount>decimal</Amount>
              </Detail>
            </Charge>
          </Transactions>
        </RTServiceTransactions>
      </RT_Customer>
    </Property>
  </ResidentTransactions>
</MRIRMUtilityCreate>
```

MITS 2.0 UtilityCreate Response

The UtilityCreate API responses for MITS 2.0 and 3.0 are identical with the exception of the header tags. For more information, refer to “MITS 3.0 UtilityCreate Response” on page 95.

3.0—`<MRIRMUtilityCreateMITS30Response Status="Pass">`

2.0—`<MRIRMUtilityCreateResponse Status="Pass">`

Defining Custom APIs 6

This chapter is intended for MRI users who are working with the Flexibility Toolkit to customize their business process within MRI.

Defined Read and Marked Read Schema

This schema provides for the definition of complex, table-driven read and marked read Web Services interfaces. Most of the standard interfaces are provided through the use of services defined using this schema.

```
<interfaceDefinition>
  <interfaceName/>            The name of the interface(required)
  <interfaceVersion/>         The version of the interface
  <applicationId/>            Application association of the
                              interface
  <interfaceType/>            Values: Defined Marked Read, Marked
                              Read
  <interfaceDescription/>     Description of the interface
  <interfaceStatus/>          Values: Active, Inactive
  <documentTag/>              The document tag to be used in the
                              response. If not provided, the
                              interface name will be used.
  <parameters>
    <parameterOverlay>
      <overlayDescription/>   Description of overlay
      <parameterOverlayNo/>   Number of this parameter overlay
      <parameter>
        <parameterTag/>       Parameter tag name
        <parameterType/>      Values: Date, String, Numeric
      </parameter>
    …
    </parameterOverlay>
    …
  </parameters>
  <segmentDefinition>
    <segmentNo/>              Assigned Segment Number
    <baseSegment/>            Segment to join against. 0 for base
                              segments.
    <segmentType/>            Values: MRI Table, Constant, Custom
                              Query
    <tableName/>              For MRI Table type, the table to be
                              read
    <selectQuery/>            For Custom Query type, the SQL Query
                              to be executed
    <segmentTag/>             Tag to be used for this segment
    <segmentOptional/>        Y indicates to ignore if table does
                              not exist or errors
    <segmentJoin>
      <joinComponent>
        <overlayLink/>        Link to determine when this join
                              should be included
```

```
        <tagName/>            Field in this segment to be joined
                              against
        <joinOperator/>       Values: =, GT, LT, GE, LE, NE, NU (is
                              NULL), BW(beginning with), EW(ending
                              with), CONTAINS
        <joinParameter/>      Request Parameter value to be used in
                              the join
      <joinBaseSegTag/>       Base Segment Tag to be used in the
                              join
        <joinValue/>          Constant value to be used in the join
        <joinValueType/>
      </joinComponent>
      …
    </segmentJoin>
    <blockorIncludeTagList/> Values: Block, Include, Include plus
                              Custom
    <tagList>
      <tagName/>              Field name
      …
    </tagList>
    <mappedTags>
      <mappedTag>
        <tagName/>            Field name to be mapped
        <newName/>            New field tag name in response
      </mappedTag>
      …
    </mappedTags>
    <mappedValues>
      <mappedValue>
        <tagName/>            Field name to be mapped
        <tagValue/>           Field value to be mapped
        <newValue/>           New mapped value
      </mappedValue>
      …
    </mappedValues>
    <additionalTags>
      <additionalTag>
        <tagName/>            Tag name to be added to the segment
                              output
        <tagValue/>           Tag value for additional tag
        <tagParam/>           Request parameter to use as tag value
      </additionalTag>
      …
    </additionalTags>
    <markings>                Section only for Defined Marked Read
      <marking>
       <markType/>            Values: Constant, Interface Id, Parameter
        <markField/>          Database field name to be updated
       <markConstant/>        Constant value to update the markField
        <markParameter/>      Request parameter to update the
                              markField
      </marking>
      …
    </markings>
  </segmentDefinition>
  …
</interfaceDefinition>
```

Generic Update Schema

This schema provides for the definition of simple, table-driven update interfaces.

```
<interfaceDefinition>
  <interfaceName/>                 The name of the interface
                                   (required)
  <interfaceVersion/>              The version of the interface
  <applicationId/>                 Application association of the
                                   interface
  <interfaceType/>                 Values: Defined Update
  <interfaceDescription/>          Description of the interface
  <interfaceStatus/>               Values: Active, Inactive
  <requestSegmentTag/>
  <targetTable/>                   Name of table to be updated
  <treatAsBatch>Y</treatAsBatch>   Values: Y/N  defaults to Y
  <allowUpdate>Y</allowUpdate>     Values: Y/N defaults to N
  <allowCreate>Y</allowCreate>     Values: Y/N defaults to N
  <generatedValues>
    <generatedValue>
      <fieldname/>                 Field name where value will be
                                   placed
      <valueType/>                 Values: MRISeq
      <sequenceValue/>             If MRISeq, SEQID in the MRISeq
                                   table
      <zeroFilled/>                Values: Y/N defaults to N
    </generatedValue>
    …
  </generatedValues>
  <staticValues>
    <staticValue>
      <tagName/>                   Tag name of static value field
      <tagValue/>                  Static value to be used for all
                                   rows
    </staticValue>
    …
  </inputStaticValues>
  <defaultValues>
    <defaultValue>
      <tagName/>                   Tag name of data to be defaulted
      <tagValue/>                  Default value
    </defaultValue>
    …
  </defaultValues>
  <additionalFieldValidations>
    <validation>
      <fieldname/>                 Field name of data to be
                                   validated
      <validationId/>              Values: GACCAccount, Period
    </validation>
    …
  </additionalFieldValidations>
  <blockedDBDefaults>
    <fieldname/>                   Field name where db default
                                   value is to be blocked
    …
```

```
  </blockedDBDefaults>
  <mappedTags>
    <mappedTag>
      <tagName/>                Tag name to be mapped
      <newName/>                New tag name to be used in
                                update
    </mappedTag>
    …
  </mappedTags>
</interfaceDefinition>
```

Stored Procedure Schema

This schema provides for the definition of stored procedure calls defined as Web Services interfaces.

```
<interfaceDefinition>
  <interfaceName />                 The name of the interface
                                    (required)
  <interfaceVersion />              The version of the interface
  <interfaceType />                 Value: Stored Procedure
  <interfaceStatus />               Values: Active, Inactive
  <interfaceDescription />          Description of the interface
  <storedProcedureName />           Stored Procedure Name
  <parameters>
    <parameter>
      <parameterTag /               Parameter Name in request
      <spParameterName />           SQL SP Parameter Name
      <spParameterDBType />         SQL Data Types
      <spParameterDirection />      Values: INPUT/OUTPUT
      <spParameterSize />
    </parameter>
    …
  </parameters>
</interfaceDefinition>
```

Custom Object Schema

This schema provides for the definition of custom object-based Web Services interfaces. This allows the use of non-MRI Software development built objects through the API Service standard Web Services interface.

```
<interfaceDefinition>
  <interfaceName/>          The name of the interface (required)
  <interfaceVersion/>       The version of the interface
  <applicationId/>          Application association of the
                            interface
  <interfaceType/>          Values: Custom Object
  <interfaceDescription/>   Description of the interface
  <interfaceStatus/>        Values: Active, Inactive
```

```
  <objectName/>              Equals the Active X Project Name and
                             the class name sperated by a period,
                             such as testObject.clsTest
</interfaceDefinition>
```

The class must have to have a public method implemented called **bProcessXMLInterface** and be defined as shown below:

```
Public Function bProcessXMLInterface(ByVal sClientId As String, _
                                     ByVal sDatabase As String, _
                                     ByVal sUserId As String, _
                                     ByVal sRequest As String, _
                                     ByRef sResponse As String, _
                                     ByRef sErrResponse As String)
                                                      As Boolean
```

Note

- **sClientId** is the client ID provided by the XML interface component and used for the authentication of this user.
- **sDatabase** is the database name provided by the XML interface component and used for authentication of this user and the target database of all database activity during the execution of this method.
- **sUserId** is the user ID provided by the XML interface component and authenticated for this method.
- **sRequest** is a string provided by the XML interface component. This string is unprocessed and will be in whatever format is provided to the Web Service.
- **sResponse** is a string that allows the return of information from the called object and class. All responses will be returned within an XML node named with the interface name plus the **Response** constant unless the **wrapResponse** flag is set to **N**.

  Example

  <yourInterfaceResponse>returned sResponse</yourInterfaceResponse>

- **sErrResponse** is a string to allow the return of error information from the called object and class. All responses will be returned within an XML node that is named with the interface name plus the **Error** constant, unless the **wrapErrResponse** flag is set to **N**.

  Example

  <yourInterfaceError>returned sErrResponse</yourInterfaceError>

Stored Procedure Method Usage

The defined stored procedure capability is provided as a way to extend the functionality provided through the standard interface methods and defined capabilities.

The following stored procedure is provided as an example. Note that the stored procedure includes the following elements:

- Input parameter (**@VendId**)
- Output parameter (**@LastInvoiceMessage**)
- Returned values from a dataset (**SELECT * FROM VEND**)

Example

```
CREATE PROCEDURE customsp_VendorInfo
(@VendId varchar(12),
 @LastInvoiceMessage varchar(40) OUTPUT)
AS
BEGIN
  SELECT * FROM VEND WHERE VENDID = @VendId

  SELECT @LastInvoiceMessage =
    'Last Invoice Dated' + CONVERT(varchar(15),
      (SELECT MAX(INVCDATE) FROM INVC WHERE VENDID =@VendId),
      107)
  RETURN
```

Web Services API Definition Schema

The schema used to create the Web Services API link to this stored procedure is shown below and is stored in the XML Interface Definitions table (TB_MRI_WSDEF). Access to this Web Services API method needs to be granted to the target user through the Security and System Administration Console.

```
<interfaceDefinition name="customVendorInfoRead">
  <interfaceName>customVendorInfoRead</interfaceName>
  <interfaceVersion>1.0</interfaceVersion>
  <interfaceType>Stored Procedure</interfaceType>
  <interfaceStatus>Active</interfaceStatus>
  <interfaceDescription />
  <applicationId />
  <storedProcedureName>customsp_VendorInfo</storedProcedureName>
  <parameters>
    <parameter>
      <spParameterOrder>1</spParameterOrder>
      <parameterTag>VENDID</parameterTag>
      <spParameterName>VENDID</spParameterName>
      <spParameterDBType>varchar</spParameterDBType>
      <spParameterSize>12</spParameterSize>
      <spParameterDirection>INPUT</spParameterDirection>
    </parameter>
```

```
    <parameter>
      <spParameterOrder>2</spParameterOrder>
      <parameterTag>LastInvoiceMessage</parameterTag>
      <spParameterName>LastInvoiceMessage</spParameterName>
       <spParameterDBType>varchar</spParameterDBType>
       <spParameterSize>40</spParameterSize>
       <spParameterDirection>OUTPUT</spParameterDirection>
    </parameter>
  </parameters>
</interfaceDefinition>
```

API Response

The following example shows the response provided from a successful call to the custom Web Services method. The dataset returned from the **SELECT * FROM VEND** portion is returned in the **NewDataSet** structure. If more than one vendor was returned, the **Table** node will be repeated. Output parameter **LastInvoiceMessage** is returned under the **SP Result** node.

Example

```
<NewDataSet>
  <Table>
    <VENDID>1DA111</VENDID>
    <VENDNME1>Big Business</VENDNME1>
    <VENDADDR>123 Main Street</VENDADDR>
    <CITY>Anytown</CITY>
    <STATE>OH</STATE>
    <ZIP>12345</ZIP>
    <FEDIDTYP>1</FEDIDTYP>
    <FEDID>999999999</FEDID>
    <M_1099REQ>N</M_1099REQ>
    <USEDISC>Y</USEDISC>
    <DELFLAG>N</DELFLAG>
    <DAYS1>5</DAYS1>
    <DISC1>5.0000</DISC1>
    <DAYS2>30</DAYS2>
    <LASTDATE>2007-01-26T14:31:44-05:00</LASTDATE>
    <USERID>WEBUSER1</USERID>
    <FOREIGNVND>N</FOREIGNVND>
    <PHONE>4405551212</PHONE>
    <STATUS>A</STATUS>
    <PAYTYPE>V</PAYTYPE>
    <COUNTY>ABC</COUNTY>
    <ENTITYRESTR>I</ENTITYRESTR>
    <WIRETOCM>N</WIRETOCM>
    <HOMEONLY>N</HOMEONLY>
    <SUBCONT>N</SUBCONT>
    <CERTREQD>N</CERTREQD>
    <SUBWITH>N</SUBWITH>
    <PRIORITYID>@</PRIORITYID>
    <ATTNFEE>N</ATTNFEE>
    <NOPOOK>N</NOPOOK>
  </Table>
</NewDataSet>
```

```
<SPResult>
  <OutParam>
    <NAME>LastInvoiceMessage</NAME>
    <VALUE>Last Invoice Dated May 02, 2006</VALUE>
  </OutParam>
</SPResult>
```

Error Codes

A

Number	Error Constant	Error Message
0	InvalidXML	Provided request parameter is not valid XML.
1	MissingTable	The requested table [table name] does not exist.
2	MissingField	The requested field [field name] does not exist in table [table name].
3	IntParseFailure	The value for field [field name] in table [table name] is not an integer.
4	DoubleParseFailure	The value for field [field name] in table [table name] is not a floating point number.
5	DateTimeParseFailure	The value for field [field name] in table [table name] is not a date.
6	InvalidLengthFailure	The value for field [field name] in table [table name] is too long.
7	MissingRequiredField	The value for field [field name] in table [table name] is required but is not provided.
8	CustomFieldUnsupportedTable	The requested table [table name] is not supported in this API.
9	AuthenticationFailed	Authentication failed. A valid session could not be retrieved.
100	kMsgGeneral	[Variable messages returned]
110	kMsgNotAuthorized	User not authorized for access the requested API. [API Name]
120	kMsgAuthenticationFailed	Authentication failed.
130	KMsgDBConnectFailed	Database Connection Failed.
140	kMsgInterfaceDefNotFound	API Interface method definition not found.
150	kMsgInvalidXMLRequest	Provided request parameter is not valid XML

Number	Error Constant	Error Message
1000	kMsgDefinedAPIGeneral	[Variable messages returned]
1010	kMsgAPIDefNotFound	API Defintion not found ([API Name]).
1020	kMsgAPIUnrecognized	Unrecognized API or type ([API Name]/[API Type]).
1030	kMsgAPIDefInvalidAppl	Invalid Application [Application ID] in the specified database ([Database Name])
1040	kMsgAPINameInconsistent	The API Name tag in the apiInfo parameter does not match the base tag of the commandInfo parameter.
1100	kMsgDefinedReadGeneral	[Variable messages returned]
1110	kMsgDRUnknownSegType	Unknown segment type ([Segment Type]).
1120	kMsgDRTableNotFound	Requested table ([Table Name]) does not exist in the database.
1130	kMsgDRFieldNotFound	Requested field ([Table Name]/[Field Name]) does not exist in the database table.
1140	kMsgDRInvalidJoinType	Invalid join type or definition.
1150	kMsgDRJoinError	Invalid value type for join.
1160	kMsgDRMarkParameterNotFound	Marking parameter value not found in request.
1170	kMsgDRSQLCommitError	Error on commit of SQL Transaction.
1175	kMsgDRSQLMarkerUpdateError	Error on updating marker fields. Records were not marked.
1180	kMsgDRInvalidSQL	Invalid SQL statement.
1190	kMsgDRInvalidParameterType	Provided parameter ([Parameter Name]=[Parameter Value]) is not of the correct type.
1191	kMsgDRInvalidParameterTypeDate	Provided parameter ([Parameter Name]=[Parameter Value]) is not of the correct type date or date/time.

Number	Error Constant	Error Message
1192	kMsgDRInvalidParameterTypeNumeric	Provided parameter ([Parameter Name]=[Parameter Value]) is not a numeric type.
1193	kMsgDRInvalidParameterTypeText	Provided parameter text ([Parameter Name]=[Parameter Value]) is longer than the allowed length.
1200	kMsgDRInvalidParameterSet	Invalid parameter set provided.
1400	kMsgDefinedUpdateGeneral	[Variable messages returned]
1410	kMsgDUSQLCommitError	Error on commit of SQL Transaction.
1420	kMsgDUSQLFailed	SQL [1] statement failed! (SQL=[SQL Statement]).
1430	kMsgDUUniqueKeyNotFound	Unble to identify unique key value.
1440	kMsgDUReqdFieldValueMissing	[Field Name] is a required field and no value has been provided.
1450	kMsgDUInvalidValue	The value [Value] is an invalid value for field [Field Name].
1460	kMsgDUAddlValidationError	Validation error.
1470	kMsgDUInvalidUpdate	Record already exists for this key value and update is not allowed.
1480	kMsgDUInvalidInsert	Record does not exist for this key value and insert is not allowed.
1490	kMsgDUValueToLong	Field value for field [1] is too long.
1500	kMsgDUValueNotDate	Field value is not a valid date ([1]).
1510	kMsgDUValueNotNumeric	Field value is not numeric ([1]).
1520	kMsgDUWrongDocTag	The request contains the incorrect document tag.
1530	kMsgDUValueNotBit	Field value is not valid for type bit ([1]).
1600	kMsgDefinedSPGeneral	[Variable messages returned]
1610	kMsgSPDuplicateParamOrder	Duplicate order numbers in SP definition.

Number	Error Constant	Error Message
1800	kMsgDefinedCreateObjGeneral	[Variable messages returned]
1810	kMsgMultiObjDefined	Multiple object name elements found in the interface definition.
1820	kMsgNoObjInDef	Object name element not found in interface definition.
1830	kMsgCreateObjError	Could not create object [Object Name].
2000	kMsgAPInvCreateGeneral	[Variable messages returned]
2010	kMsgAPInvCreateInvalidRequestXML	The provided string for Creating Invoices is not valid XML.
2020	kMsgAPInvCreateInvalidResponseXML	The returned string from Create Invoices is not valid XML.
2030	kMsgAPInvCreateNoInterfaceId	Unable to generate an InterfaceId for use in the Create Invoice process.
2040	kMsgAPInvCreateInvalidAppl	Invalid Application AP in the specified database.
2050	kMsgAPInvCreateNoInvoices	No invoices found in the request XML.
3000	kMsgCMExpGeneral	[Variable messages returned]
3010	kMsgCMExpInvalidParameters	Invalid parameter set provided.
3020	kMsgCMExpInvalidAppl	Invalid Application CM in the specified database.
4000	kMsgGLBudReadGeneral	[Variable messages returned]
4010	kMsgGLBudReadMissingReqdFields	Request is missing required fields.
4020	kMsgGLBudReadInvalidRequestXML	The provided request string is not valid XML.
4030	kMsgGLBudReadInvalidAppl	Invalid Application GL in the specified database.
4040	kMsgGLBudReadInvalidYearEndPd	Invalid Budget Year Ending Period provided.
4500	kMsgGLBudUpGeneral	[Variable messages returned]

Number	Error Constant	Error Message
4510	kMsgGLBudUpDeleteWhereError	Error occured in building the Where clause to delete existing budget records.
4520	kMsgGLBudUpLockError	Unable to overwrite existing locked budget records without proper authorization.
4530	kMsgGLBudUpCreateFailed	Creation of Budget records failed.
4540	kMsgGLBudUpDeleteExistingFailed	Error occured in deleting the existing budget information.
4550	kMsgGLBudUpNoBudgetsFound	No budget information found in the request XML.
4560	kMsgGLBudUpInvalidRequestXML	The string provided for the request is not valid XML.
4570	kMsgGLBudUpInvalidApp	Invalid Application GL in the specified database.
4580	kMsgGLBudUpMissingInfo	Insufficient information provided to create budget records (BUDTYPE, ENTITYID, BASIS, DEPT, YEARENDINGPD)
4590	kMsgGLBudUpBudgetInsertError	Error occured on insert to the Budgets (BUDGETS) table.
4600	kMsgGLBudUpBudgetNoteInsertError	Error occured on insert to the Budget Note (BUDNOTE) table.
4610	kMsgGLBudUpErrorOnCommit	Error on commit of SQL Transaction.
4620	kMsgGLBudUpInvalidBudType	Invalid Budget Type provided.
4630	kMsgGLBudUpInvalidEntityId	Invalid Entity ID provided.
4640	kMsgGLBudUpInvalidBasis	Invalid Basis code provided.
4650	kMsgGLBudUpInvalidDept	Invalid Department provided.
4660	kMsgGLBudUpInvalidYearEndingPd	Invalid Year Ending Period provided.
4670	kMsgGLBudUpInvalidAcctNum	Invalid Account Number provided.
4680	kMsgGLBudUpInvalidBudNoteType	Invalid Budget Note Type provided.

Number	Error Constant	Error Message
4690	kMsgGLBudUpInvalidBudNoteBegPrd	Invalid Budget Note Beginning Period provided.
4700	kMsgGLBudUpInvalidBudNoteEndPrd	Invalid Budget Note Ending Period provided.
5000	kMsgGLJrnlCreateGeneral	[Variable messages returned]
5010	kMsgGLJrnlCreateMissingReqd	Required field value missing for [Field Name].
5020	kMsgGLJrnlCreateInvalidValue	Invalid value in field [Field Name] ([Field Value]).
5030	kMsgGLJrnlCreateSQLInsFailed	Error occured during SQL Insert attempt.
5040	kMsgGLJrnlCreatePostPriorFailed	Error occured on attempt to Post Prior Period/Year entries.
5050	kMsgGLJrnlCreateSQLCommitError	Error on commit of SQL Transaction.
5060	kMsgGLJrnlCreateInvalidRequest	The provided request string is not valid XML.
5070	kMsgGLJrnlInvalidPeriod	Invalid Financial Period value provided.
5200	kMsgRMGuestCardCreateError	Error Occured during Guest Card creation.
5210	kMsgRMGuestCardCreateNoneFound	No Guest Card entries found in the request.
5300	kMsgRMResTranInvalidAppl	Invalid RM Application in the specified database.
5310	kMsgRMResTranCreatePost	Error occurred on attempt to post Resident Transactions.
5320	kMsgRMResTranReadInvalidRequestXML	The provided string for Creating Resident Transactions is not valid XML.
5330	kMsgRMResTranReadInvalidResponseXML	The returned string from Create Transactions is not valid XML.
5400	kMsgRMResCustomerReadInvalidAppl	Invalid RM Application in the specified database.

Number	Error Constant	Error Message
5410	kMsgRMResCustomerReadProcess	Error occurred on attempt to Read Customer Information.
5420	kMsgRMResCustomerReadInvalidRequestXML	The provided string for retrieving Customer Information is not valid XML.
5430	kMsgRMResCustomerReadInvalidResponseXML	The return string from Create Customer Information is not valid XML.
5500	kMsgRMResEditMissingElement	Required field value missing for [Field Name].