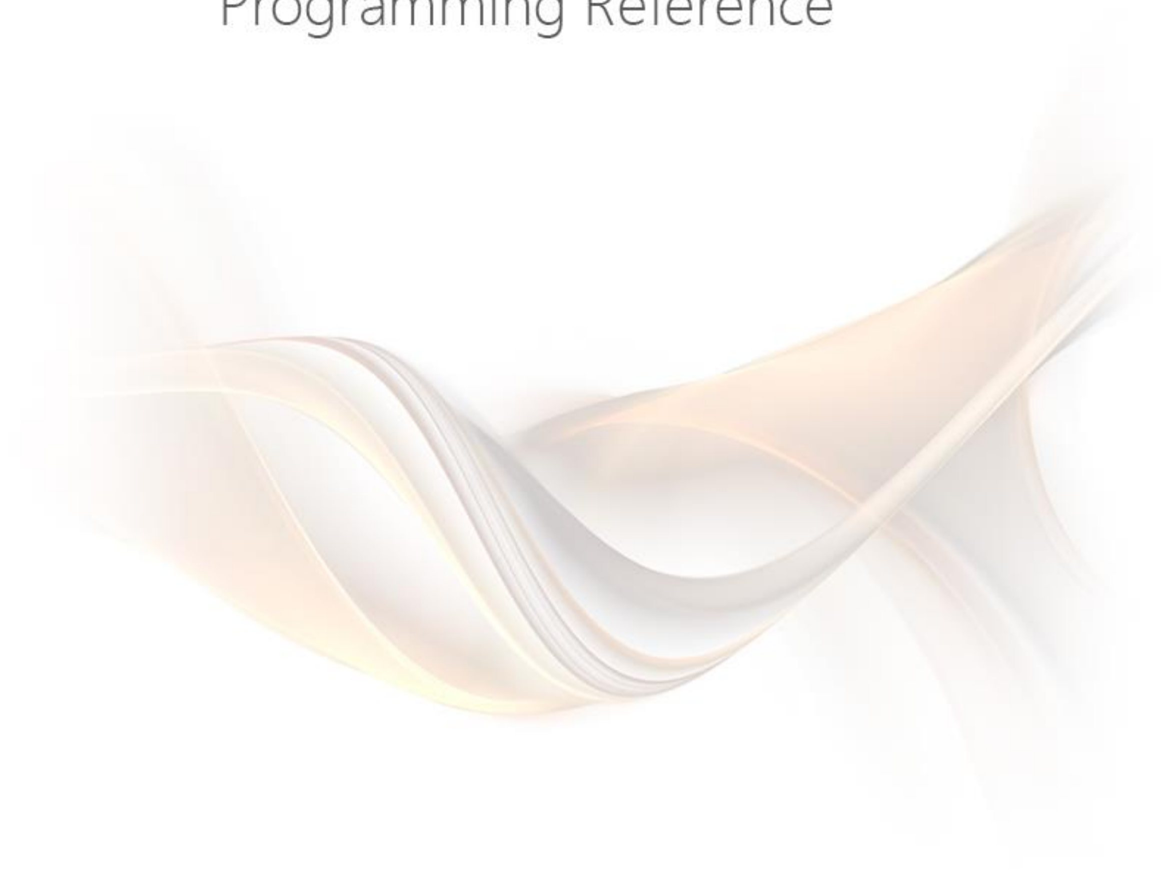
Visions Enterprise 5

Visions API

Programming Reference



Version 5.5 U100

Contents

General

Overview

Several application program interfaces (API) are available to allow you to perform certain actions without using the Visions Enterprise user interface (UI). This allows third party developers to use web service calls to get or update Visions data.

The Visions APIs are divided into functional areas:

- **General**: allowing you to query both site and user names for accurate use in the calls
- **Assets**: allowing you to manage equipment level actions; such as creation, static data updates, linking, or data retrieval
- **Work Memos**: allowing you to manage work memos, both inspection tasks and work requests; such as creation, update, or data retrieval
- **Dashboard**: allowing you to get dashboard counts for both the user and general dashboards
- **TML**: allowing you to manage thickness monitoring locations and readings; such as creating, updating, or querying TML identifiers, readings, and notes data
- **CP Testing**: allowing you to manage cathodic protection test points and readings; such as creating, updating, or querying CP Test Point identifiers, readings, and notes data
- **General Trending**: allowing you to manage general trending test points and readings; such as creating, updating, or querying trend point identifiers, readings, and trend types data

Access

Navigation

The web service calls are expressed as a C# interface, with the individual calls documented in the applicable section.

```
public interface IVisAPI {
...
}
```

Reference the applicable chapter for the available calls and their requirements, as well as an example for each call.

Conditions

1. Each API is available as a separate module; which you must purchase for the calls you wish to access.
   a. For calls pertaining to asset management, the Asset API module must be active.
   b. For calls pertaining to work memo management, the Work Memo API module must be active.

   c. For calls pertaining to querying dashboard results, the Dashboard API module must be active.

   d. For calls pertaining to the TML management, the TML API module must be active.

   e. For calls pertaining to the CP Testing management, the CP Testing API module must be active.

   f. For calls pertaining to the General Trending management, the General Trending API module must be active.

   Note: Each API may have specific business conditions and rules, please reference the appropriate chapter for all details.

2. All returned data will be based on the Plant and Equipment Type security rights defined for the Visions User name by which you connect.

3. Each module is grouped and provides the following routines within the API, as follows:

   a. General:

      i. QuerySites

      ii. QueryUsers

   b. Asset API:

      i. CreateEquipment

      ii. UpdateEquipment

      iii. MoveEquipment

      iv. RenumberEquipment

      v. DeleteEquipment

      vi. RemoveEquipment

      vii. QueryEquipment

      viii. EquipmentExistsQuery

      ix. UpdateSchedule

      x. EquipmentListQuery

      xi. CreateLink

      xii. UpdateLink

      xiii. CheckForLink

      xiv. RemoveLink

   Reference the Assets chapter for full details.

   c. Work Memo API:

      i. CreateWorkOrder

ii. UpdateWorkOrder

iii. QueryWorkOrder

iv. WorkOrderExistsQuery

v. MemoListQuery

vi. WorkRequestListQuery

vii. InspectionTaskListQuery

Reference the Work Memos chapter for full details.

d. Visions API - Dashboards

i. QueryDashboard

ii. QueryUserDashboard

iii. QueryDashboardList

Reference the Dashboards chapter for full details.

e. Visions API - TMLs

i. QueryTML

ii. CreateTML

iii. CloneTML

iv. CloneTMLWithDates

v. UpdateTML

vi. DeleteTML

vii. AddTMLReadings

viii. EditTMLReadings

ix. DeleteTMLReadings

x. AddTMLNotes

Reference the TML chapter for full details.

f. Visions API - CP Testing

i. QueryCPTP

ii. CreateCPTP

iii. CloneCPTP

iv. UpdateCPTP

v. DeleteCPTP

vi. AddCPTPReadings

vii. EditCPTPReadings

viii. DeletCPTPReadings

ix. AddCPTPNotes

Reference the CP Testing chapter for full details.

g. Visions API - General Trending

i. QueryGTTP

ii. QueryGTTrendTypes

iii. CreateGTTrendTypes

iv. UpdateGTTrendTypes

v. DeleteGTTrendTypes

vi. CreateGTTP

vii. UpdateGTTP

viii. DeleteGTTP

ix. AddGTTPReadings

x. EditGTTPReadings

xi. DeleteGTTPReadings

Reference the General Trending chapter for full details.

Input

Each of the APIs provide access to the following calls:

- **QuerySites**: query to return a list of the available site names
- **QueryUsers**: query the list of available users

Calls

The structure, input and output variables and datatypes, for each call within each API.

```
string[] QuerySites();

bool QueryUsers(string SiteName, string UserName, string Password, out string UserList,
out string ErrorMsg);
```

Calling Conventions

The web service is a standard SOAP web service and may be consumed by anything that can consume SOAP web services. The following conventions are generic to all available APIs.

- All parameter types are intentionally kept to simple types to aid in technical access to the service by other platforms and avoid complex protocol errors with XML format which is why field values are passed as simple XML rather than arrays of strings or structs.
- Every API call returns a boolean value – true if successful or false if an error occurred.

- Every API call takes an ErrorMsg as an out string parameter, which is populated with the error message if the call fails.
- API calls which reference a piece of equipment do so by its three (3) logical keys – PlantNum, EquipType, and EquipNum – not our internal Visions account identifier.
- Excluding a parameter means that the area will not be filtered. For example, in the query calls if you do not wish to include specific values to filter the Plants or Equipment Types, simply exclude the PlantNum and EquipType field name placeholders and values.
- There is a switch, <DebugInfo />, which can be included in the Values parameter to cause the SOAP replies to contain additional trace information useful to debugging.
- API calls to interact with existing memos reference them by the dual logical keys of MemoType and MemoNo, which together uniquely identify any work memo with values accessible to the user.
- Arbitrary field values may be specified using the Values parameter, passing one or more field values as XML in a manner detailed below.
- Any operations the API performs are performed with the account its connection parameters (SiteName, UserName, Password) specify, and ensure that user access restrictions are respected by the asset registry.

  [SiteName]: represents the database name as would be used in the credentials of the Visions Login

  [Username]: represents the user name as would be used in the credentials of the Visions Login

  [Password]: represents the password for the specified user name, as would be used in the credentials of the Visions Login
- The password is sent over the line as plain text in the SOAP message.
- The API web service ensures that the user account has the correct functional security permissions to perform the associated operation.
- The API service will check whether the appropriate API Module has been purchased before allowing any calls through. There are currently three planned API modules – DashboardAPI, AssetAPI and WorkLogAPI – which may be purchased separately.
- You may use square brackets instead of <> for nested HTML tags in Values. This is useful for editing your calls in a tool like SoapUI. If you do not use them for [Values], then normal HTML tags are used and square brackets won't interfere with XML encoding.
- The URL for the web service's Basic HTTP Endpoint is:

  http://[servername]:7137/VisAPI/VisAPI_BH
- The URL to retrieve WSDL for the web service is as follows:

  http://[servername]:7137/VisAPI?wsdl

  where [ServerName] is the name of your Visions application server (aka middle tier)

Behavior Notes

- The API web service will correct the casing of any values that already exist in the Visions database (i.e., EquipNum, PlantNum, etc.) unless doing so would introduce ambiguity (i.e., conflicting records differing only in case exist), in which case it reads the values case-sensitively.

- The record creation API calls create a record in the associated parent table (StatHdr or WorkLog), as well as zero or more child records in child tables (i.e., DesMat, BoilerStatic, etc.) needed to store values specified in the Values XML string.
- Fields supporting the Visions lookup lists will be validated and any values that do not exist in the field's associated list will be passed back as an error within the response.
- No libraries or special components are necessary to consume the API web service.

List Parameters and Data Format

Bool: boolean data types supports various values as valid:

True: T, True, 1, Y, Yes

False: F, False, 0, N, No

FieldValues: allow multiple list queries; each of which may take up to four fields in the form of a field name and list of permitted values

Note: All records matching any of the given values in each field are returned.

The list queries return two-dimensional string arrays which duplicate the grid indexes shown in the Visions Client. Row zero of the returned array has the column headers, and each row thereafter is a data row. By default, the returned columns directly mirror the relevant log in the Visions client, but they may be changed or limited by providing a ColumnSet.

ColumnSet: a filter string which may be applied to a list of data classes in order to organize its hierarchical data into a flat view suitable for display in a 2D grid control or any other GUI element with a similar topography; allows you to format the response dataset using the foreign references and field names to join related tables within a List Query call in an API, as well as assign an alias for your foreign reference(s); for example, to return only the logical keys for equipment, specify this ColumnSet value: "PlantNum,EquipNum,EquipType"

MapID: Metegrity can provide custom integration mappings that alter the default names of Visions fields to match and provide some limited logical rules for re-interpreting the data being sent into the API. In some cases, clients do not want to use our fields – they want to send their own data in and have it translated by our API routines into our own fields. In that case, you load a provided custom API mapping using the Visions Database Configuration utility and specify the name of it in the MapID parameter to any of the API calls. If you do not have a custom mapping, leave the MapID field null and the call will function as documented.

Standard MapID option(s):

ReturnRTF: an option to control whether memo (rich text format supported) field values will return the string value or the RTF exactly as formatted within the database.

- The flag must explicitly be set within a call to return the rich text formatting (RTF) within the value(s) of all string (Memo) fields within a call

- If the flag is used; the RTF source including the transcoded binary form of the embedded object will be returned
- If the flag is not used; then the placeholder string “(Embedded OLE Object)” will be returned as the value for the field by the API rather than trying to strip a megabyte-or-larger string in a way that consumes vast amounts of server memory.

  Note: RTF supported fields are identified in the Reference Fields with a data type of “string (Memo)”.

Limitation: If a memo field contains both text and an embedded object, such as an image, when the RTF flag is not used, none of the text will be returned only the place holder string will be returned.

Reference the List Query call examples in each API chapter for more information.

XML Field Values

The API calls accept values for arbitrary fields in a very simple XML format, illustrated below:

```
<Values>
  <Value field='EquipName' >EX-01</Value>
  <Value field='rfDesMat.MAWP' >560</Value>
  <Value field='rfSchedule.DueDate'
        fmt='MM-dd-yyyy'>08-16-2012</Value>
</Values>
```

This XML could be sent to create a new piece of equipment with the specified Equipment Name, MAWP, and Due Date.

The only significant tag is “Value”, and the only significant attributes are “field” and “fmt”. The field attribute is specified in the Tables and Fields group of References section in each API chapter. This allows the specification of field values in child tables such as DesMat or Schedule when equipment is created or updated through the asset registry API.

All date fields require that a format string be included (in the “fmt” attribute) with their value, so that we are able to interpret the value correctly and pre-emptively avoid any issues with date formatting, time zones, locale settings and so forth. The syntax of the date format string is the .NET standard, and is described here:

http://msdn.microsoft.com/en-us/library/az4se3k1

and here:

http://msdn.microsoft.com/en-us/library/8kb3ddd4

Validation and Error Management

Each of the API calls will generate a response logging any validation or data errors within the call. Examples of some of the more common items:

1. Specified equipment record 'D80-HC-22/U-3000 - Demonstration DB/Boiler' already exists.

Note: This identifies that equipment being added already exists in Visions.

2. Error: Field 'Outage Extent' is required but not provided.

Note: This identifies that a value for a required attribute was not provided in the call.

3. Input string was not in a correct format.

Note: This identifies that the value for an attribute was not provided in the correct format, such as: a decimal where whole number was expected.

4. Cannot locate property 'OperatingLimi' of dataclass 'DesMat' specified in column set.

Note: This identifies that the attribute name provided in the call does not actually exist. In this case it was a spelling mistake where the attribute name was supposed to be “OperatingLimit”.

Call Examples

QuerySites

Allows you to get a list of the available sites defined within a database. These are the same sites as defined within the Visions Application Server (middle tier) Database Configuration utility.

Call

```
<!-- This call returns a list of configured sites in the current database.  It is the
    only call that doesn't require login parameters, and can be used to provide users
    with a drop-down list for Site selection on logging into something integrated with
    Visions. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:QuerySites/>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

The response shows that the two available site names are: Production and Testing.

```
<!-- Returned sites are listed in CSV format. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <QuerySitesResponse xmlns="http://metegrity.com">
    <QuerySitesResult xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
       xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
       <a:string>Production,Training</a:string>
    </QuerySitesResult>
   </QuerySitesResponse>
 </s:Body>
</s:Envelope>
```

QueryUsers

Allows you to get a list of the available user account names

Call

```
<!-- This call returns a list of user accounts and associated full names for the
    indicated site. It can be used to provide input to QueryUserDashboard, allowing the
    desired user to be selected from a list. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:QueryUsers>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
   </met:QueryUsers>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

The response shows the list of user account names and their associated full names.

```
<!-- Results are returned in a CSV list, formatted as "UserName1, FullName1, UserName2,
    FullName2, UserName3..." -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <QueryUsersResponse xmlns="http://metegrity.com">
    <QueryUsersResult>true</QueryUsersResult>
    <UserList>jsmith,John Smith,mdonovan,Mary Donovan,bwaters,Bob
   Waters,VIS_SERVICE,Visions Service User</UserList>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </QueryUsersResponse>
 </s:Body>
</s:Envelope>
```

ReturnRTF

Allows you to get the full RTF (rich text format) from the database for all string(Memo) fields listed in the call.

RTF-Enabled Call

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:met="http://metegrity.com"
xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
  <soapenv:Header/>
  <soapenv:Body>
    <met:MemoListQuery>
      <met:SiteName>[Visions SiteName]</met:SiteName>
      <met:UserName>[Visions UserName]</met:UserName>
      <met:Password>[Visions Password]</met:Password>
      <met:FieldName1>IntegrityCritical</met:FieldName1>
      <met:FieldValues1>
        <arr:string>0</arr:string>
      </met:FieldValues1>
      <met:FieldName2>MemoType</met:FieldName2>
      <met:FieldValues2>
        <arr:string>WR</arr:string>
      </met:FieldValues2>
      <met:ColumnSet>rfEquipment=>E;rfSuperSubLink_EquipmentID=>SSL;rfMajorEquipment=>ME;
       E.PlantNum,E.EquipType,E.EquipNum,E.SSL.ME.EquipNum:Associated_Major,MemoType,
       MemoNo,DateCreated,MemoTitle,Completed,CloseFlag,ActivityCode,
       FullPartFlag,WorkOrderNo,PriorityCode,DateCompleted,
       DueDate,AssignedUser,ProbDesc,WorkDesc,ProposedDisp</met:ColumnSet>
      <met:MapID>ReturnRTF</met:MapID>
    </met:MemoListQuery>
  </soapenv:Body>
</soapenv:Envelope>
```

RTF-Enabled Response

The response shows the RTF within the string(Memo) field listed in the call.

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <MemoListQueryResponse xmlns="http://metegrity.com">
      <MemoListQueryResult>true</MemoListQueryResult>
      <OutList xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
         <a:ArrayOfstring>
            <a:string>Plant ID</a:string>
            <a:string>Equipment Type</a:string>
            <a:string>Equipment Number</a:string>
            <a:string>Associated Major</a:string>
            <a:string>Type</a:string>
            <a:string>Memo No</a:string>
            <a:string>Date Created</a:string>
            <a:string>Title</a:string>
            <a:string>Completed</a:string>
            <a:string>Status</a:string>
            <a:string>Activity Code</a:string>
            <a:string>Extent</a:string>
            <a:string>Work Order No.</a:string>
            <a:string>Priority Code</a:string>
            <a:string>Date Completed</a:string>
            <a:string>Due Date</a:string>
            <a:string>Assigned User</a:string>
            <a:string>Problem Desc</a:string>
            <a:string>Work Desc</a:string>
            <a:string>Proposed Disp</a:string>
         </a:ArrayOfstring>
         <a:ArrayOfstring>
            <a:string>U-3000 - Demonstration DB</a:string>
            <a:string>PSV</a:string>
            <a:string>D08-PSV-100</a:string>
            <a:string i:nil="true"/>
            <a:string>WR</a:string>
            <a:string>122</a:string>
            <a:string>20/12/2007</a:string>
            <a:string>Inspection Access</a:string>
            <a:string>True</a:string>
            <a:string>Completed</a:string>
            <a:string>TA-SW</a:string>
            <a:string>Full</a:string>
            <a:string>000</a:string>
            <a:string>High</a:string>
            <a:string>02/05/2012</a:string>
            <a:string>02/05/2012</a:string>
            <a:string i:nil="true"/>
            <a:string/>
               <a:string>{\rtf1\ansi\deff0{\fonttbl{\f0\fnil\fcharset0 Arial;}{\f1\fnil
Arial;}{\f2\fnil\fcharset2 Symbol;}}\viewkind4\uc1\pard\lang4105\fs20 Select bull plugs
to be removed as
follows:\par\par\pard{\pntext\f2\'B7\tab}{\*\pn\pnlvlblt\pnf2\pnindent0{\pntxtb\'B7}}\fi-
720\li720 Two columns, two plugs wide, one at each end of both header plugsheets.\f1\par
\f0{\pntext\f2\'B7\tab}One column, two plugs wide, at the center of both header
plugsheets.\f1\par
}</a:string>
```

```
        <a:string/>
    </a:ArrayOfstring>
```

Additional Information

Reference each individual API chapter.

Assets API

Overview

Several application program interfaces (API) are available to allow you to perform certain actions without using the Visions Enterprise user interface (UI). This allows third party developers to use web service calls to get or update Visions data.

The Visions API for assets allows you to manage equipment level actions; such as creation, static data updates, linking, or getting data.

Access

Navigation

The web service calls are expressed as a C# interface, with the individual calls, conventions, references, and examples documented below.

```
public interface IVisAPI {
...
}
```

Conditions

1. The Asset API is a separate module which must be active to use its calls.
2. Assets management through the API for the Foreign Reference tables rely on the modules that are licensed within the Visions site to which you are connecting, and are managed through the equipment types.
   a. StatHdr, MiscStat, Schedule, and DesMat are part of the Base module, and only allow equipment types derived from Circuit and Misc Custom.
   b. BoilerStatic, CTStatic, FilterStat, FTBoilStat, FrStatic, StackStat, TankStatic and Vessel are part of the Pressure Vessel module, and only allow equipment types derived from Boiler, Cooling Tower, Filter, Firetube Boiler, Furnace, Stack, Tank, and Vessel.
   c. BundleStatic, ExStatic, and PltExchStat are part of the Exchanger module, and only allow equipment types derived from Bundle, Exchanger, and Plate Exchanger.
   d. CPStat is part of the Advanced Cathodic Protection module, and only allow equipment types derived from CP System, CP Rectifier, and CP Misc.
   e. InjPntStat, PipeStat, and ValveStat are part of the Piping module, and only allow equipment types derived from Injection Point, Pipe, and Valve.
   f. PL_SegmentStat and PL_WellheadStat are part of the Pipeline module, and only allow equipment types derived from Pipeline Misc, Pipeline Segment, and Wellhead.

   g. RE_GPTurbStat, RE_MotorStat, and RE_PumpStat are part of the Rotating Equipment module, and only allow equipment types derived from Centrifugal Pump, GP Steam Turbine, and RE Motor.

3. Link Class EquipComp (4) which allows components to be linked to assets rely on the modules that are licensed within the Visions site to which you are connecting, and are managed through the component types.
   a. Component types derived from Misc Custom are part of the Base module.
   b. Component Types derived from Critical Valve, Expansion Joint, Hanger, and Pipe are part of the Piping module.
4. Link Class DriverDriven (8) which allows drivers and driven to be linked to rotating equipment assets rely on the modules that are licensed within the Visions site to which you are connecting, and are managed through the equipment types.
   a. Equipment Types derived from Centrifugal Pump, GP Steam Turbine, or RE Motor are part of the Rotating Equipment module.
5. Link Class CPLinkRect (12) which allows rectifiers to be linked to CP System assets rely on the modules that are licensed within the Visions site to which you are connecting, and are managed through the equipment types.
   a. Equipment Types derived from CP Rectifier or CP System are part of the Advanced CP module.
6. Only equipment derived from the Circuit base type can be specified as input to CircuitEquipQuery. Any other equipment specified will return an error from the equipment type validator.
7. The asset hierarchy (Business Units, Plants, Equipment Types, and Component Types) must first be configured in Visions to manage assets.
8. Security to perform calls are dependent upon the security access for the username passed into a call. For example: if you do not have Move Equipment privileges, you will not be able to call the MoveEquipment call successfully. Security functions for Assets are:
   a. **Add or Clone Equipment**: controls asset creation and cloning (equivalent to New or Clone Equipment within the Visions Client)
   b. **Change Equipment Type**: controls change of the type of equipment for assets; Equipment Type Rights for the username used within the call also control which types of equipment are permitted
   c. **Delete Equipment**: controls deletion of assets
   d. **Move Equipment**: controls change of a plant and/or train/unit for assets; Plant Rights for the username used within the call also control which Plants are permitted
   e. **Renumber Equipment**: controls change of the equipment number for assets
   f. **Change Equipment Due Date**: controls rescheduling assets (equipment's Next Full Inspection/Service Due Date)

g. **Delete/Restore in Recycle Bin**: controls permanent deletion of assets

h. **Add or Change Plants**: controls editing the Business Unit for a Plant (when Link Class = UnitPlant within the Linking API calls)

i. **Link Plant Circuits**: controls editing the Circuit(s) for a Plant (when Link Class = CircuitPlant within the Linking API calls)

9. Lookup List items validate against the lookup lists bound to the applicable static data form, table and field.

Input

The Asset API provides the following calls:

◢ **CreateEquipment**: create assets, simulates the 'New Equipment' and 'New Circuit' actions on the Equipment Index which allows you to add new assets to Visions

Note: On all new assets, the recycle bin flag defaults to "No" so all assets are active.

Security: Permission is dependent upon the 'Add or Clone Equipment' function.

◢ **UpdateEquipment**: update existing assets, simulates the update to much of the Static Data information for assets

Security: Permission is dependent upon the 'Edit Static Data/Component Datasheets' function.

◢ **MoveEquipment**: change the plant for asset(s), simulates the 'Move' action on the Equipment Index which allows you to change the Plant and/or Train/Unit for one or more assets

Security: Permission is dependent upon the 'Move Equipment' function.

◢ **RenumberEquipment**: change the equipment number for asset(s), simulates the 'Renumber' action on the Equipment Index which allows you to change the Equipment Number for an asset

Security: Permission is dependent upon the 'Renumber Equipment' function.

◢ **DeleteEquipment**: set the recycle bin flag for asset(s); simulates the 'Delete' action on the Equipment Index which allows you to delete one or more assets, placing them in the Recycle Bin

Security: Permission is dependent upon the 'Delete Equipment' function.

◢ **RemoveEquipment**: permanently remove asset(s); simulates the 'Delete Permanently' action in the Recycle Bin which permanently deletes an asset and all of its related data

Security: Permission is dependent upon the 'Delete/Restore in Recycle Bin' function.

◢ **QueryEquipment**: query the assets, returning the value(s) for the attribute specified within the call

◢ **EquipmentExistsQuery**: query the assets, verifying whether the asset(s) exist

◢ **UpdateSchedule**: change the due date for an asset; simulates 'Reschedule' on the Scheduling

Security: Permission is dependent upon the 'Change Equipment Due Date' function.

- **EquipmentListQuery**: query the assets, returning the attributes from the based on the requested attributes
- **CreateLink**: create one or more links; the type of which is specified by the LinkClass parameter; simulates the various linking grids within the Static Data forms, such as Circuits, Services, PSV Protection, Equipment Linking, etc.

  Reference the Linking API > Classes of Links section for permissions pertaining to each link class.
- **UpdateLink**: update an existing link; the type of which is specified by the LinkClass parameter
- **CheckForLink**: query the links, returning whether the link exists; the type of which is specified by the LinkClass parameter
- **RemoveLink**: remove an existing link; the type of which is specified by the LinkClass parameter
- **CircuitEquipQuery**: select a list of circuits as input, returning all the (non-Circuit) equipment linked to those circuits as output

Reference the Linking API section below for details on the Link Class parameters.

Calls

The structure, input and output variables and datatypes, for each call within the Asset API.

```
bool CreateEquipment(string SiteName, string UserName, string Password, string EquipNum,
    string PlantNum, string EquipType, string Values, out string ErrorMsg);

bool DeleteEquipment(string SiteName, string UserName, string Password, string EquipNum,
    string PlantNum, string EquipType, out string ErrorMsg);

bool EquipmentExistsQuery(string SiteName, string UserName, string Password, string
    EquipNum, string PlantNum, string EquipType, out bool existsResult, out string
    ErrorMsg);

bool EquipmentListQuery(string SiteName, string UserName, string Password, string
    FieldName1, string[] FieldValues1, string FieldName2, string[] FieldValues2, string
    FieldName3, string[] FieldValues3, string FieldName4, string[] FieldValues4, string
    DateFmt, string ColumnSet, int pageSize, int pageNum, out string[][] OutList, out bool
    moreLeft, out string ErrorMsg);

bool MoveEquipment(string SiteName, string UserName, string Password, string EquipNum,
    string PlantNum, string EquipType, string NewPlantNum, string NewTrainNum, out string
    ErrorMsg);

bool QueryEquipment(string SiteName, string UserName, string Password, string EquipNum,
    string PlantNum, string EquipType, string FieldPath, out string FieldVal, out string
    ErrorMsg);

bool RenumberEquipment(string SiteName, string UserName, string Password, string
    EquipNum, string PlantNum, string EquipType, string NewEquipNum, out string ErrorMsg);
```

```
bool RemoveEquipment(string SiteName, string UserName, string Password, string EquipNum,
    string PlantNum, string EquipType, out string ErrorMsg);

bool UpdateEquipment(string SiteName, string UserName, string Password, string EquipNum,
    string PlantNum, string EquipType, string Values, out string ErrorMsg);

bool UpdateSchedule(string SiteName, string UserName, string Password, string EquipNum,
    string PlantNum, string EquipType, string DueDate, string DateFmt, out string
    ErrorMsg);

bool CreateLink(string SiteName, string UserName, string Password, LinkClass Class,
    string Values,  string KeyA1, string KeyA2, string KeyA3, string KeyB1, string KeyB2,
    string KeyB3, out string ErrorMsg);

bool UpdateLink(string SiteName, string UserName, string Password, LinkClass Class,
    string Values,  string KeyA1, string KeyA2, string KeyA3, string KeyB1, string KeyB2,
    string KeyB3, out string ErrorMsg);

bool CheckForLink(string SiteName, string UserName, string Password, LinkClass Class,
    string TypeOfLink,  string KeyA1, string KeyA2, string KeyA3, string KeyB1, string
    KeyB2, string KeyB3, out string ErrorMsg, out bool IsLinkPresent);

bool RemoveLink(string SiteName, string UserName, string Password, LinkClass Class,
    string KeyA1, string KeyA2, string KeyA3, string KeyB1, string KeyB2, string KeyB3,
    out string ErrorMsg);

bool CircuitEquipQuery(string SiteName, string UserName, string Password, string
    CircuitNum, string PlantNum, string EquipType, string FieldName1, string[]
    FieldValues1, string FieldName2, string[] FieldValues2, string FieldName3, string[]
    FieldValues3, string DateFmt, string ColumnSet, int pageSize, int pageNum, out
    string[][] OutList, out bool moreLeft, out string ErrorMsg, string MapID, string
    Values);
```

Calling Conventions

The web service is a standard SOAP web service and may be consumed by anything that can consume SOAP web services.

The following conventions are specific to the Asset API:

- API calls which reference a piece of equipment do so by its three (3) logical keys – PlantNum, EquipType, and EquipNum – not our internal Visions account identifier.
- The Recycle Bin flag (DeleteFlag) is implicitly assumed to be false; the API does not give access to the recycle bin at present.
- Arbitrary field values may be specified using the Values parameter, passing one or more field values as XML in a manner detailed below.
- The QueryEquipment call returns a single field value from the specified piece of equipment as a string, which may be parsed as desired.
- The API web service ensures that the user account has the correct functional security permissions to perform the associated operation, (i.e., Add Equipment, Delete Equipment, Rename Equipment, or Move Equipment).

- The EquipmentExistsQuery has two boolean return values – the function returns false if the call fails for some reason (i.e., connectivity), but returns true and sets “existsResult” to true or false based on whether the item exists otherwise.
- The CircuitEquipQuery by default uses its own ColumnSet which includes the fields (Plant ID, Train / Unit, Equipment Type, Equipment Number, Circuit Plant, Circuit Type, Circuit Number and Avail on IR) when ColumnSet is [null]. However, the user can specify additional fields to return or provide their own custom ColumnSet normally.
- You may create more than one piece of equipment with a single call to the API. To do so, you omit the key fields from the parameters in the function call (or send null), and include them in Values. You then include multiple <Values> tags inside a <Records> tag.

  Note: Example shown below in the Example Calls section.
- The URL for the web service’s Basic HTTP Endpoint is:

  http://[servername]:7137/VisAPI/VisAPI_BH
- The URL to retrieve WSDL for the web service is as follows:

  http://[servername]:7137/VisAPI?wsdl

  Where [ServerName] is the name of your Visions application server (aka middle tier)

Behavior Notes

- The API web service will correct the casing of any values that already exist in the Visions database (i.e., EquipNum, PlantNum, etc.) unless doing so would introduce ambiguity (i.e., conflicting records differing only in case exist), in which case it reads the values case-sensitively.
- The record creation API calls create a record in the associated parent table (StatHdr or WorkLog), as well as zero or more child records in child tables (i.e., DesMat, BoilerStatic, etc.) needed to store values specified in the Values XML string.
- DeleteEquipment sets the Recycle Bin flag, placing the equipment in the recycle bin; conversely, RemoveEquipment removes it entirely.
- The UpdateSchedule call may be duplicated by the UpdateEquipment call using Values in the case where the 3rd-party developer cannot make the .NET DateTime data type function on their end; it is included for simplicity of use only.
- MoveEquipment changes the plant or train (or both) associated with a piece of equipment. If null values are passed for either, the field remains as it was prior to the call. MoveEquipment breaks the links of any Linked Minor Equipment before changing its plant, and moves any associated Minor equipment when it is called to move a piece of Major Equipment. Equipment can be moved to a new Business Unit by moving it to a plant located in the BU in question.
- The CircuitEquipQuery has the following:
  - A single circuit can be specified using the three key fields PlantNum, EquipType and CircuitNum (i.e., EquipNum). These values together uniquely identify a Circuit.

- PlantNum and/or EquipType can be omitted if the EquipNum unambiguously identifies a specific circuit in the database. If it does not, or matches a non-Circuit, then a user-friendly error message will be returned.
- The parameters FieldName1/2/3 and FieldValues1/2/3 can be used to narrow down the circuited equipment returned; these fields operate exactly as the similarly-named fields in the **EquipmentListQuery** API call do. (Thus, for example, the user could retrieve only all the Pipe-type equipment in a given circuit.)

◢ In order to change any key value within one of the Linking API calls, you must first delete the existing record using RemoveLink, and then recreate the link with the new keys using CreateLink.

List Parameters and Data Format

FieldValues: allow multiple list queries; each of which may take up to four fields in the form of a field name and list of permitted values

Note: All records matching any of the given values in each field are returned.

The list queries return two-dimensional string arrays which duplicate the grid indexes shown in the Visions Client. Row zero of the returned array has the column headers, and each row thereafter is a data row. By default, the returned columns directly mirror the relevant log in the Visions client, but they may be changed or limited by providing a ColumnSet.

ColumnSet: allows you to format the response dataset using the foreign references and field names to join related tables within a List Query call in an API, as well as assign an alias for your foreign reference(s); for example, to return only the logical keys for equipment, specify this ColumnSet value: “PlantNum,EquipNum,EquipType”

Reference the EquipmentListQuery call example for more information.

XML Field Values

The API calls accept values for arbitrary fields in a very simple XML format, illustrated below and in the examples section:

```
<Values>
<Value field='EquipName' >EX-01</Value>
<Value field='rfDesMat.MAWP' >560</Value>
<Value field='rfSchedule.DueDate'
 fmt='MM-dd-yyyy'>08-16-2012</Value>
</Values>
```

This could be sent to create a new piece of equipment with the specified Equipment Name, MAWP and Due Date.

The only significant tag is “Value”, and the only significant attributes are “field” and “fmt”. The field attribute is specified in the Tables and Fields group of References section below. This allows the specification of field values in child tables such as DesMat or Schedule when equipment is created or updated through the Asset API.

All date fields require that a format string be included (in the “fmt” attribute) with their value, so that we are able to interpret the value correctly and pre-emptively avoid any issues with date formatting, time zones, locale settings and so forth. The syntax of the date format string is the .NET standard, and is described here:

http://msdn.microsoft.com/en-us/library/az4se3k1

and here:

http://msdn.microsoft.com/en-us/library/8kb3ddd4

The same type of date format string is included as a parameter to the UpdateSchedule API call.

Linking API

As part of the Asset API, specific calls are available to simulate the linking done within the Static Data for an asset.

Security: To manage links, permission is dependent upon the user having rights to the Equipment Types defined within the site (schema).

These calls are: CreateLink, UpdateLink, CheckForLink and RemoveLink.

The LinkClass parameter is an enumeration, determining the form of link in question:

```
enum LinkClass {
    EquipLink       = 1,
    EquipCircuit    = 2,
    CircuitPlant    = 3,
    EquipComp       = 4,
    Protection      = 5,
    MajorMinor      = 6,
    DriverDriven    = 7,
    UnitPlant       = 8,
    PlantTrain      = 9,
    EquipService    = 10
    CPProtection    = 11
    CPRectLink      = 12
}
```

Based on the specified LinkClass, the “key” parameters are interpreted differently as described below. They are always the logical key fields of the two entities in the link, however.

Limitation: You cannot use ‘UpdateLink’ to change any of the key values, you must first delete the link and recreate it.

Classes of Links

The following list describes each class within the linking API calls, including the key requirements and any additional field values that are available.

EquipLink

A link between two pieces of equipment which does not match any of the more specific types listed below.

- The TypeOfLink parameter provides a text description of the link's purpose, and is part of the logical key for the link record
- KeyA1, KeyA2 and KeyA3 contains the PlantNum, EquipType and EquipNum for the first piece of equipment
- KeyB1, KeyB2 and KeyB3 contains the PlantNum, EquipType and EquipNum of the second piece of equipment
- All links of the EquipLink class are reciprocal, existing in both directions simultaneously
- To specify the type of link, use the Values string with the following:

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	TypeOfLink	("Type of Link", dtString, Maximum Length 15)
		The type or reason the equipment is linked.

EquipCircuit

A link between a conventional piece of equipment and a circuit, specifying that the equipment is part of that circuit.

- KeyA1, KeyA2 and KeyA3 contains the PlantNum, EquipType and EquipNum of the equipment
- KeyB1, KeyB2 and KeyB3 contains the PlantNum, EquipType and EquipNum of the circuit
- To specify whether the equipment will be available on a circuit's integrity report, use the Values string with the following:

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
bool	API570Insp	("Available on Integrity Report", dtInteger, Default 0)
		Indicates whether the circuited equipment requires API570 inspection documentation.

CircuitPlant

Associates a circuit with a plant

Note: A circuit must be linked to a plant before it can be linked to equipment within that plant.

Security: Permission is dependent upon the 'Link Plant Circuits' function.

- KeyA1, KeyA2 and KeyA3 contains the PlantNum, EquipType and EquipNum for the circuit
- KeyB1 contains the PlantNum for the plant ID
- KeyB2 and KeyB3 are not in use

EquipComp

Associates a component with a piece of equipment.

Limitation: A component can only be linked to one piece of equipment at a time.

- KeyA1, KeyA2 and KeyA3 contains the PlantNum, EquipType and EquipNum for the equipment
- KeyB1, KeyB2 and KeyB3 contains the PlantNum, CompType and ComponentNum for the component
- To specify the link date and/or location for a component, use the Values string with the following:

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
DateTime?	LinkDate	("Link Date", dtDate)
		The date the component was linked to the equipment.
	Location	("Location", dtString, Maximum Length 35)
		The location where the component was linked to the equipment.

CPProtection

Associates a cathodic protection (CP) equipment with the target equipment that it protects.

- KeyA1, KeyA2 and KeyA3 contains the PlantNum, EquipType and EquipNum for the non-CP equipment being protected
- KeyB1, KeyB2 and KeyB3 contains the PlantNum, EquipType and EquipNum for the CP equipment that does the protecting
- To specify the type of link, use the Values string with the following:

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	TypeOfLink	("Type of Link", dtString, Maximum Length 15)
		The type or reason the equipment is linked.

- To conform with the Visions applications, the following validators are in place:

Supported Protected Equipment Types	Supported Protecting Equipment Types
Boiler, Bundle, Cooling Tower, Exchanger, Filter, Firetube Boiler, Furnace, Injection Point, Misc Custom, Pipe, Pipeline Custom, Pipeline Segment, Plate Exchanger, PSV, PSV Location, ST Topside, ST Fixed Underwater, ST Helipad, ST Crane, Stack, Tank, Vessel, Wellhead	CP System, CP Custom, CP Rectifier

Limitation: Only listed Equipment Types have the CP Protection grid available.

Protection

Associates a PSV or similar device with the equipment it protects.

- KeyA1, KeyA2 and KeyA3 contains the PlantNum, EquipType and EquipNum for the equipment being protected

- KeyB1, KeyB2 and KeyB3 contains the PlantNum, EquipType and EquipNum for the PSV that does the protecting
- To specify the type of link, use the Values string with the following:

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	TypeOfLink	("Type of Link", dtString, Maximum Length 15)
		The type or reason the equipment is linked.

- To conform with the Visions applications, the following validators are in place:

Supported Protected Equipment Types	Supported Protecting Equipment Types
Boiler, Bundle, Exchanger, Filter, Firetube Boiler, Furnace, Pipe, Tank, Vessel, Misc Custom, Injection Point, Stack, Pipeline Segment, Wellhead, Pipeline Custom, Cooling Tower, Plate Exchanger, Centrifugal Pump	PSV, PSV Location

Limitation: Only listed Equipment Types have the PSV Protection grid available.

MajorMinor

Associates a piece of equipment as a Minor Equipment associated with a different piece of equipment which is Major.

Limitation: The classification for equipment that is to be linked as the Minor equipment must be “Equipment”. You cannot link equipment classified as “Circuit” or “Major” to a Major asset.

- KeyA1, KeyA2 and KeyA3 contains the PlantNum, EquipType and EquipNum for the Major equipment
- KeyB1, KeyB2 and KeyB3 contains the PlantNum, EquipType and EquipNum for the Minor Equipment

DriverDriven

Associates a driver or similar device with the equipment it powers or rotates.

- KeyA1, KeyA2 and KeyA3 contains the PlantNum, EquipType and EquipNum for the drivers
- KeyB1, KeyB2 and KeyB3 contains the PlantNum, EquipType and EquipNum for the driven equipment
- To specify the type of link, use the Values string with the following:

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	TypeOfLink	("Type of Link", dtString, Maximum Length 15)
		The type or reason the equipment is linked.

- To conform with the Visions applications, the following validators are in place:

Supported Driver Equipment Types	Supported Driven Equipment Types
RE Motor, GP Steam Turbine	Centrifugal Pump

Limitation: Only listed Equipment Types have the Drivers/Driven grid available.

UnitPlant

Allows you to change the unit to which a plant belongs.

Security: Permission is dependent upon the 'Add or Change Plants' function.

- KeyA1 contains the UnitID
- KeyB1 contains the PlantNum
- KeyA2, KeyA3, KeyB2 and KeyB3 are not used

PlantTrain

Defines a train for a given plant; allows you to add a train/unit to a plant, or change plant ownership for a train/unit

Security: Permission is dependent upon the 'Add or Change Plants' function.

- KeyA1 contains the PlantNum
- KeyB1 contains the TrainNum
- KeyA2, KeyA3, KeyB2 and KeyB3 are not used
- To manage all information for a train/unit, use the Values string with any of the following:

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	Description	("Description", dtString, Maximum Length 200)
		A brief description for a train/unit.
string	Notes	("Notes", dtString, Maximum Length 4000)
		Full descriptive notes for a train/unit.
string	PlotPlan	("Plot Plan", dtString, Maximum Length 255)
		The plot plan drawing for a train/unit. Must contain the full source path and filename and is restricted to '.dwg' files.

EquipService

Associates a service with a piece of equipment.

- KeyA1, KeyA2 and KeyA3 contains the PlantNum, EquipType and EquipNum for the equipment
- KeyB1 and KeyB2 contains the ServiceID, ShellorTube for the service
- KeyB3 is not used
- To conform with the Visions applications, the following validators are in place:

ShellOrTube Options	Supported Supplied Equipment Type
Waterside, Fireside	Boiler, Firetube Boiler
Tubeside	Bundle, Pipe, PL Segment, PL Custom, Valve, Wellhead
Shellside	Cooling Tower, Filter, Misc Custom, Tank, Vessel, Centrifugal Pump, GP Steam Turbine
Shellside, Tubeside	Exchanger, Furnace, Injection Point, Plate Exchanger, PSV, PSV Location, Stack

Limitation: Only listed Equipment Types have the Services grid available.

CPLinkRect

Associates a CP rectifier with a CP system to represent an impressed current system.

- KeyA1, KeyA2 and KeyA3 contains the PlantNum, EquipType and EquipNum for the CP System equipment
- KeyB1, KeyB2 and KeyB3 contains the PlantNum, EquipType and EquipNum for the CP Rectifier equipment
- To conform with the Visions applications, the following validators are in place:

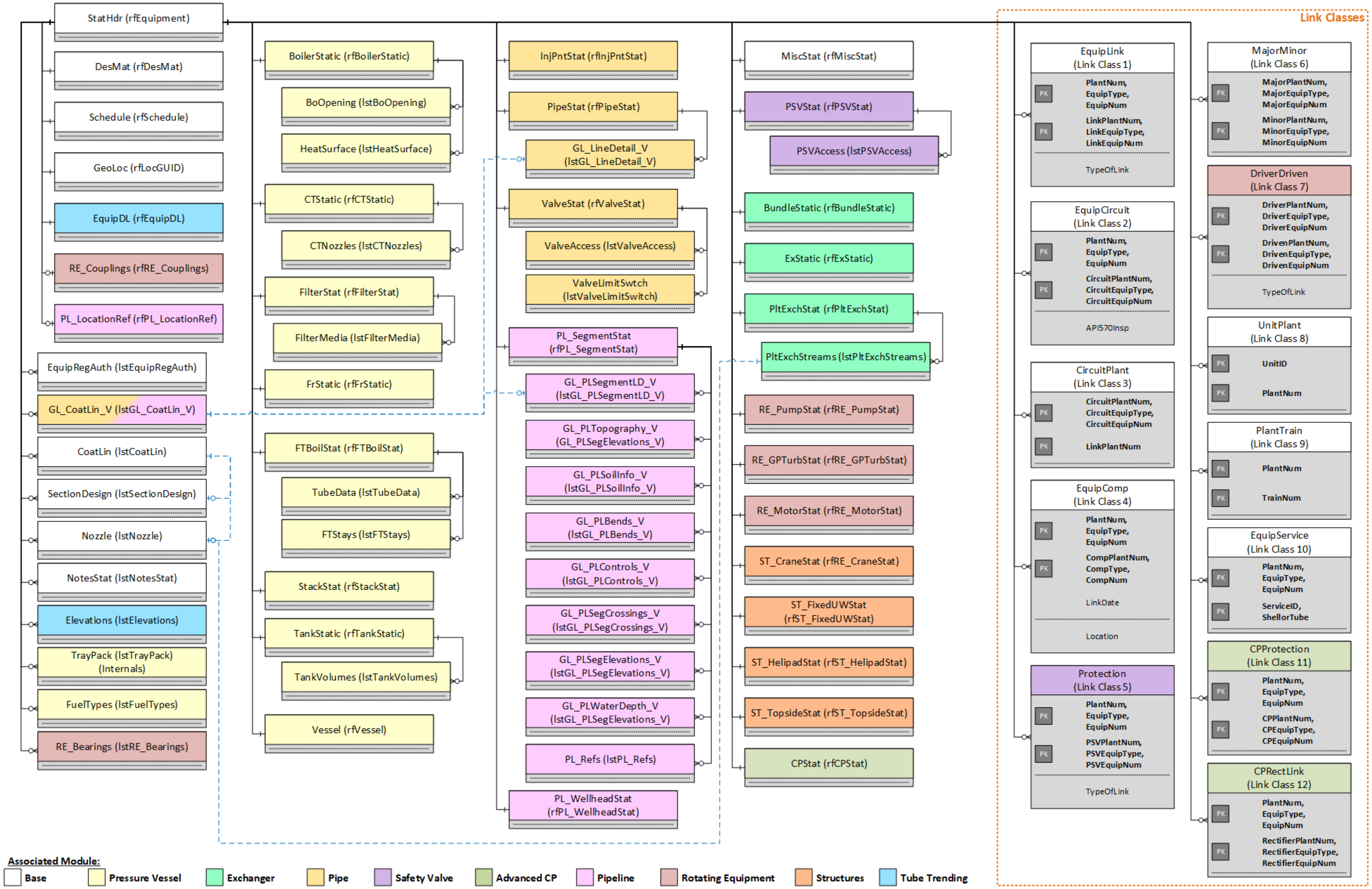
Supported CP System Equipment Type	Supported Rectifier Equipment Type
CP System	CP Rectifier

Limitation: Only listed Equipment Types have the CP Protection grid available.

References

Data Model Overview

Reference the following Data Model Overview page for an ERD of the supported tables for the Asset API.


StatHdr (rfEquipment)
DesMat (rfDesMat)
Schedule (rfSchedule)
GeoLoc (rfLocGUID)
EquipDL (rfEquipDL)
RE_Couplings (rfRE_Couplings)
PL_LocationRef (rfPL_LocationRef)
EquipRegAuth (lstEquipRegAuth)
GL_CoatLin_V (lstGL_CoatLin_V)
CoatLin (lstCoatLin)
SectionDesign (lstSectionDesign)
Nozzle (lstNozzle)
NotesStat (lstNotesStat)
Elevations (lstElevations)
TrayPack (lstTrayPack) (Internals)
FuelTypes (lstFuelTypes)
RE_Bearings (lstRE_Bearings)
BoilerStatic (rfBoilerStatic)
BoOpening (lstBoOpening)
HeatSurface (lstHeatSurface)
CTStatic (rfCTStatic)
CTNozzles (lstCTNozzles)
FilterStat (rfFilterStat)
FilterMedia (lstFilterMedia)
FrStatic (rfFrStatic)
FTBoilStat (rfFTBoilStat)
TubeData (lstTubeData)
FTStays (lstFTStays)
StackStat (rfStackStat)
TankStatic (rfTankStatic)
TankVolumes (lstTankVolumes)
Vessel (rfVessel)
InjPntStat (rfInjPntStat)
PipeStat (rfPipeStat)
GL_LineDetail_V (lstGL_LineDetail_V)
ValveStat (rfValveStat)
ValveAccess (lstValveAccess)
ValveLimitSwtch (lstValveLimitSwitch)
PL_SegmentStat (rfPL_SegmentStat)
GL_PLSegmentLD_V (lstGL_PLSegmentLD_V)
GL_PLTopography_V (GL_PLSegElevations_V)
GL_PLSoilInfo_V (lstGL_PLSoilInfo_V)
GL_PLBends_V (lstGL_PLBends_V)
GL_PLControls_V (lstGL_PLControls_V)
GL_PLSegCrossings_V (lstGL_PLSegCrossings_V)
GL_PLSegElevations_V (lstGL_PLSegElevations_V)
GL_PLWaterDepth_V (lstGL_PLSegElevations_V)
PL_Refs (lstPL_Refs)
PL_WellheadStat (rfPL_WellheadStat)
MiscStat (rfMiscStat)
PSVStat (rfPSVStat)
PSVAccess (lstPSVAccess)
BundleStatic (rfBundleStatic)
ExStatic (rfExStatic)
PltExchStat (rfPltExchStat)
PltExchStreams (lstPltExchStreams)
RE_PumpStat (rfRE_PumpStat)
RE_GPTurbStat (rfRE_GPTurbStat)
RE_MotorStat (rfRE_MotorStat)
ST_CraneStat (rfRE_CraneStat)
ST_FixedUWStat (rfST_FixedUWStat)
ST_HelipadStat (rfST_HelipadStat)
ST_TopsideStat (rfST_TopsideStat)
CPStat (rfCPStat)
Link Classes
EquipLink (Link Class 1)
PK PlantNum, EquipType, EquipNum
PK LinkPlantNum, LinkEquipType, LinkEquipNum
TypeOfLink
EquipCircuit (Link Class 2)
PK PlantNum, EquipType, EquipNum
PK CircuitPlantNum, CircuitEquipType, CircuitEquipNum
API570Insp
CircuitPlant (Link Class 3)
PK CircuitPlantNum, CircuitEquipType, CircuitEquipNum
PK LinkPlantNum
EquipComp (Link Class 4)
PK PlantNum, EquipType, EquipNum
PK CompPlantNum, CompType, CompNum
LinkDate
Location
Protection (Link Class 5)
PK PlantNum, EquipType, EquipNum
PK PSVPlantNum, PSVEquipType, PSVEquipNum
TypeOfLink
MajorMinor (Link Class 6)
PK MajorPlantNum, MajorEquipType, MajorEquipNum
PK MinorPlantNum, MinorEquipType, MinorEquipNum
DriverDriven (Link Class 7)
PK DriverPlantNum, DriverEquipType, DriverEquipNum
PK DrivenPlantNum, DrivenEquipType, DrivenEquipNum
TypeOfLink
UnitPlant (Link Class 8)
PK UnitID
PK PlantNum
PlantTrain (Link Class 9)
PK PlantNum
PK TrainNum
EquipService (Link Class 10)
PK PlantNum, EquipType, EquipNum
PK ServiceID, ShellorTube
CPProtection (Link Class 11)
PK PlantNum, EquipType, EquipNum
PK CPPlantNum, CPEquipType, CPEquipNum
CPRectLink (Link Class 12)
PK PlantNum, EquipType, EquipNum
PK RectifierPlantNum, RectifierEquipType, RectifierEquipNum
Associated Module:
Base
Pressure Vessel
Exchanger
Pipe
Safety Valve
Advanced CP
Pipeline
Rotating Equipment
Structures
Tube Trending

Equipment Type Tables and Link Classes

This table provides a list of the tables associated with each supported type of equipment.

Supported Equipment Type	API Table	Associated Static Data Panel	LinkClass
Boiler	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	Protection (Link Class 5)
	BoilerStatic	Design, Other	CPProtection (Link Class 11)
	BoOpening	Furnace Openings	EquipComp (Link Class 4)
	HeatSurface	Heating Surface	
	EquipRegAuth	Regulatory Authorities	
	FuelTypes	Fuel Type	
	SectionDesign	Section Design	
	Nozzle	Nozzles	
	NotesStat	Notes	
	Elevations[1]	Elevations	
	EquipDL[1]	Design	
Bundle	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	Protection (Link Class 5)
	BundleStatic	Design, Other	CPProtection (Link Class 11)
	EquipRegAuth	Regulatory Authorities	
	SectionDesign	Section Design	
	CoatLin	Coatings	
	NotesStat	Notes	
CP Custom	StatHdr	Static Data, Hazard Classification	EquipLink (Link Class 1)
	DesMat	Design, Other	CPProtection (Link Class 11)
	Schedule		
	GeoLoc	Coordinates	
	EquipRegAuth	Regulatory Authorities	
	CPStat	Design, Anode, Other	
	NotesStat	Notes	
CP Rectifier	StatHdr	Static Data, Hazard Classification	EquipLink (Link Class 1)
	DesMat	Design, Other	CPProtection (Link Class 11)
	Schedule		
	GeoLoc	Coordinates	
	EquipRegAuth	Regulatory Authorities	
	CPStat	Design, Other	
	NotesStat	Notes	
CP System	StatHdr	Static Data, Hazard Classification	CPRectLink (Link Class 12)
	DesMat	Static Data, Other	EquipLink (Link Class 1)
	Schedule		CPProtection (Link Class 11)
	GeoLoc	Coordinates	
	CPStat	Other	
	NotesStat	Notes	

Centrifugal Pump	StatHdr	Static Data, Hazard Classification	DriverDriven (Link Class 7)
	DesMat	Design, Other	EquipCircuit (Link Class 2)
	Schedule		EquipService (Link Class 10)
	GeoLoc	Coordinates	EquipLink (Link Class 1)
	EquipRegAuth	Regulatory Authorities	Protection (Link Class 5)
	RE_PumpStat	Design, Operating Conditions, Site Data, Liquid, Materials, Performance, Other	
	RE_Couplings	Couplings	
	Nozzle	Nozzles	
	RE_Bearings	Bearings	
	NotesStat	Notes	
Circuit	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design	EquipService (Link Class 10)
	Schedule		Protection (Link Class 5)
	NotesStat	Notes	
Cooling Tower	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	Protection (Link Class 5)
	EquipRegAuth	Regulatory Authorities	CPProtection (Link Class 11)
	CTStatic	Design, Basin, Other	
	CTNozzles	Spray Nozzles	
	CoatLin	Coating	
	NotesStat	Notes	
Exchanger	StatHdr	Static Data, Hazard Class	EquipCircuit (Link Class 2)
	DesMat	Design, Shellside, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	Protection (Link Class 5)
	BundleStatic	Tubeside	CPProtection (Link Class 11)
	ExStatic	TEMA, Bundle, Other	EquipComp (Link Class 4)
	EquipRegAuth	Regulatury Authorities	
	SectionDesign	Section Design	
	TrayPack	Internals	
	CoatLin	Coatings	
	NotesStat	Notes	
Filter	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	Protection (Link Class 5)
	EquipRegAuth	Regulatory Authorities	CPProtection (Link Class 11)
	FilterStatic	Design, Other	EquipComp (Link Class 4)
	FilterMedia	Filter Media	
	SectionDesign	Section Design	
	Nozzle	Nozzles	
	NotesStat	Notes	

Firetube Boiler	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	Protection (Link Class 5)
	EquipReqAuth	Regulatory Authorities	CPProtection (Link Class 11)
	FTBoilStat	Design, Other	EquipComp (Link Class 4)
	FTStays	Stays	
	TubeData	Tube Data	
	FuelTypes	Fuel Type	
	SectionDesign	Section Design	
	Nozzle	Nozzles	
	NotesStat	Notes	
Furnace	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	EquipDL[1]	Design	EquipLink (Link Class 1)
	Schedule		Protection (Link Class 5)
	GeoLoc	Coordinates	CPProtection (Link Class 11)
	EquipRegAuth	Regulatory Authorities	EquipComp (Link Class 4)
	FRStatic	Design, Other	
	FuelTypes	Fuel Types	
	SectionDesign	Section Design	
	Nozzle	Nozzles	
	Elevations[1]	Elevations	
	NotesStat	Notes	
GP Steam Turbine	StatHdr	Static Data, Hazard Classification	DriverDriven (Link Class 7)
	DesMat	Design, Other	EquipCircuit (Link Class 2)
	Schedule		EquipService (Link Class 10)
	GeoLoc	Coordinates	EquipLink (Link Class 1)
	RE_GPTurbStat	Design, Operating Conditions, Site Data, Turbine Data, Casing Design, Steam Design, Performance, Other	
	RE_Couplings	Couplings	
	NotesStat	Notes	
Injection Point	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	Protection (Link Class 5)
	EquipRegAuth	Regulatory Authorities	CPProtection (Link Class 11)
	InjPntStat	Receiving Stream, Injecting Streams, Other	
	SectionDesign	Section Design	
	Nozzle	Nozzles	
	CoatLin	Coatings	
	NotesStat	Notes	
Major	StatHdr	Static Data, Hazard Classification	EquipLink (Link Class 1)
	DesMat	Design, Other	MajorMinor (Link Class 6)
	Schedule		
	GeoLoc	Coordinates	
	EquipRegAuth	Regulatory Authorities	
	NotesStat	Notes	

Misc Custom	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	Protection (Link Class 5)
	EquipRegAuth	Regulatory Authorities	CPProtection (Link Class 11)
	MiscStat	Design, Other	EquipComp (Link Class 4)
	SectionDesign	Section Design	
	Nozzle	Nozzles	
	CoatLin	Coating	
	NotesStat	Notes	
PSV, PSV Location	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	Schedule		Protection (Link Class 5)
	GeoLoc	Coordinates	CPProtection (Link Class 11)
	EquipRegAuth	Regulatory Authorities	
	PSVStat	General, Process Conditions, Pressure and Vacuum, Flame Arrestor, Other	
	PSVAccess	Access	
	NotesStat	Notes	
Pipe	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	Protection (Link Class 5)
	EquipRegAuth	Regulatory Authorities	CPProtection (Link Class 11)
	PipeStat	Design, Other	EquipComp (Link Class 4)
	GL_LineDetail_V	Line Data	
	GL_CoatLin_V	Coating	
	NotesStat	Notes	
Pipeline Custom	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Static Data, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	CPProtection (Link Class 11)
	EquipRegAuth	Regulatory Authorities	
	PL_LocationRef	UTM Reference	
	MiscStat	Other	
	SectionDesign	Section Design	
	Nozzle	Nozzles	
	CoatLin	Coating	
	NotesStat	Notes	
	PL_StatusHistory	Status History	

Pipeline Segment	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	Protection (Link Class 5)
	EquipRegAuth	Regulatory Authorities	CPProtection (Link Class 11)
	PL_SegmentStat	Design, Emergency Planning Zone, Right of Way, Other	EquipComp (Link Class 4)
	PL_LocationRef	UTM Reference	
	PL_Refs	References	
	GL_PLSegmentLD_V	Joint/Line Data	
	GL_PLTopography_V	Topography	
	GL_SoilInfo_V	Soil	
	GL_GLControls_V	Controls	
	GL_PLBends_V	Bends	
	GL_PLSegElevations_V	Elevations	
	GL_PLWaterDepth_V	Water Depths	
	GL_CoatLin_V	Coating	
	NotesStat	Notes	
	PL_StatusHistory	Status History	
Plate Exchanger	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	Protection (Link Class 5)
	EquipRegAuth	Regulatory Authorities	CPProtection (Link Class 11)
	PltExchStat	Design, Other	
	PltExchStreams	Process Streams	
	SectionDesign	Section Design	
	Nozzle	Nozzles	
	CoatLin	Coating	
	NotesStat	Notes	
RE Motor	StatHdr	Static Data, Hazard Classification	DriverDriven (Link Class 7)
	DesMat	Design, Other	EquipLink (Link Class 1)
	Schedule		
	GeoLoc	Coordinates	
	EquipRegAuth	Regulatory Authorities	
	RE_MotorStat	Design, Site Data, Other	
	RE_Couplings	Couplings	
	RE_Bearings	Bearings	
	NotesStat	Notes	
ST Crane	StatHdr	Static Data, Hazard Classification	EquipLink (Link Class 1)
	DesMat	Design, Other	CPProtection (Link Class 11)
	Schedule		EquipComp (Link Class 4)
	GeoLoc	Coordinates	
	EquipRegAuth	Regulatory Authorities	
	ST_CraneStat	Design, Other	
	SectionDesign	Section Design	
	CoatLin	Coating	
	NotesStat	Notes	

ST Fixed Underwater	StatHdr	Static Data, Hazard Classification	EquipLink (Link Class 1)
	DesMat	Design, Other	CPProtection (Link Class 11)
	Schedule		EquipComp (Link Class 4)
	GeoLoc	Coordinates	
	EquipRegAuth	Regulatory Authorities	
	ST_FixedUWStat	Design, Other	
	SectionDesign	Section Design	
	CoatLin	Coating	
	NotesStat	Notes	
ST Helipad	StatHdr	Static Data, Hazard Classification	EquipLink (Link Class 1)
	DesMat	Design, Other	CPProtection (Link Class 11)
	Schedule		EquipComp (Link Class 4)
	GeoLoc	Coordinates	
	EquipRegAuth	Regulatory Authorities	
	ST_HelipadStat	Design, Other	
	SectionDesign	Section Design	
	CoatLin	Coating	
	NotesStat	Notes	
ST Topside	StatHdr	Static Data, Hazard Classification	EquipLink (Link Class 1)
	DesMat	Design, Other	CPProtection (Link Class 11)
	Schedule		EquipComp (Link Class 4)
	GeoLoc	Coordinates	
	EquipRegAuth	Regulatory Authorities	
	ST_TopsideStat	Design, Other	
	SectionDesign	Section Design	
	CoatLin	Coating	
	NotesStat	Notes	
Stack	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	Protection (Link Class 5)
	EquipRegAuth	Regulatory Authorities	CPProtection (Link Class 11)
	StackStat	Design, Guy Wires, Other	EquipComp (Link Class 4)
	SectionDesign	Section Design	
	Nozzle	Nozzles	
	CoatLin	Coating	
	NotesStat	Notes	
Tank	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	Protection (Link Class 5)
	EquipRegAuth	Regulatory Authorities	CPProtection (Link Class 11)
	TankStatic	Design, Other	EquipComp (Link Class 4)
	TankVolumes	Tank Volumes	
	SectionDesign	Section Design	
	Nozzle	Nozzles	
	CoatLin	Coating	
	NotesStat	Notes	

Valve	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	CPProtection (Link Class 11)
	EquipRegAuth	Regulatory Authorities	
	ValveStat	Process, Actuator, Power Gas, Pilot, Solenoid, Other	
	PSVStat	Process	
	ValveLimitSwtch	Limit Switches	
	ValveAccess	Accessories	
	NotesStat	Notes	
Vessel	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Design, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	Protection (Link Class 5)
	EquipRegAuth	Regulatory Authorities	CPProtection (Link Class 11)
	Vessel	Design, Other	EquipComp (Link Class 4)
	SectionDesign	Section Design	
	Nozzle	Nozzles	
	TrayPack	Internals	
	CoatLin	Coating	
	NotesStat	Notes	
Wellhead	StatHdr	Static Data, Hazard Classification	EquipCircuit (Link Class 2)
	DesMat	Static Data, Other	EquipService (Link Class 10)
	Schedule		EquipLink (Link Class 1)
	GeoLoc	Coordinates	Protection (Link Class 5)
	EquipRegAuth	Regulatory Authorities	CPProtection (Link Class 11)
	PL_LocationRef	UTM Reference	EquipComp (Link Class 4)
	PL_Wellhead	Static Data, Other	
	SectionDesign	Section Design	
	Nozzle	Nozzles	
	NotesStat	Notes	
	PL_StatusHistory	Status History	

[1] The Tube Trending module must be available.

Table and Field Values

The table and field structures and definitions provide the 'Values' string within a call. Fields requiring the data to be entered based on a site configured unit of measure are identified by "Unit-Based Field". Fields requiring the unit of measure value to be entered using the UoM Factors Lookup Data based on the Unit Category are identified by "UoM Factors Unit [Category]".

Limitation: Only the tables and fields listed below are supported for the Asset API.

Note: Please confirm the configurable field settings that have been set for your site to ensure that all required, read-only and default field level business rules are met within your calls. The list of potential configurable fields is listed in the Visions Administrator Help file.

StatHdr (rfEquipment)

Equipment Identification: identifies the assets defined within a site

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)

string	**UnitID**	("Unit ID", Required Field, Read-Only Field, Maximum Length 35, FK)
		The business unit where the equipment is located. Note: Validates against the configured Business Units.
string	**PlantNum**	("Plant ID", dtString, Required Field, Read-Only Field, Maximum Length 35)
		The plant where the equipment is located. Note: Validates against the configured Plants as well as the Unit ID and Plant ID combination.
string	**EquipType**	("Equipment Type", dtString, Required Field, Read-Only Field, Maximum Length 35, FK)
		The type of equipment, from the available user configured types of equipment. Note: Validates against the configured Equipment Types.
string	**EquipNum**	("Equipment Number", dtString, Required Field, Read-Only Field, Maximum Length 35)
		The asset or unique identification for the equipment.
string	ApplicationType	("Application Type", dtString, Maximum Length 35)
		From a lookup list, select the basic application of the equipment; typically identifies the orientation or configuration of the equipment. i.e.: Vertical, Horizontal for a Vessel, Heating or Recovery for a Boiler.
string	CRN	("CRN", dtString, Maximum Length 20)
		The Canadian Registration Number, only the applicable provincial code after the decimal point. This can be obtained from the ASME Data Report or the equipment nameplate.
DateTime?	DateCreated	("Date Created", dtDate, Read-only)
		The date the equipment was created. Note: Defaults to today's date on creation of the record.
DateTime?	DateDeleted	("Date Deleted", dtDate, Read-only)
		The date the equipment was flagged for deletion. Note: Defaults to today's date when the equipment is placed in the recycle bin (flagged for deletion).
bool	DeleteFlag	("Recycle Bin Flag", dtInteger, Required Field)
		Indicate whether the equipment has been flagged for deletion; defaults to "No" on creation of equipment. Choices are: 0, N, No, 1, Y, Yes.
string	DrawingNum	("Drawing Number", dtString, Maximum Length 250)
		The equipment CAD drawing.
string	EquipName	("Equipment Name", dtString, Maximum Length 200)
		The name of the equipment in a standardized format.
string	ERPNumber	("ERP Number", dtString, Maximum Length 30)
		The Enterprise Resource Planning Number. Can be used as a link between Visions and programs such as SAP, JD Edwards, PeopleSoft, etc.
string	FlowDiagram	("Flow Diagram", dtString, Maximum Length 250)
		The identification of a Process Flow Diagram which contains the equipment. This can be CAD or hard copy.
string	FuncLevel	("Functional Level", dtString, Maximum Length 25)
		The ERP functional level of the equipment.

string	FunctionalLocation	("Functional Location", dtString, Maximum Length 50)
		The ERP functional location of the equipment. Note: The functional location validates against the format of the defined mask (when set).
string	HazardClass	("Hazard Class", dtString, Maximum Length 20)
		From a lookup list, select the hazard class or category.
string	HazJustification	("Justification", dtString, Maximum Length 4000)
		The justification for the hazard class selection.
bool	InService	("In Service", dtInteger, Required Field)
		The in-service flag identifies whether the equipment is currently in or out of service. Identifies whether the equipment will be considered in calculations. Choices are: 0, N, No, 1, Y, Yes. **Note**: Defaults to "Y" (Yes, 1) unless otherwise specified.
string	Location	("Location", dtString, Maximum Length 40)
		From a lookup list, select the specific location of the equipment. This can be a building, plant and direction, grid coordinates, etc.
string	ManfDwgNum	("Mfr Drawing No", dtString, Maximum Length 250)
		The identification of the manufacturer construction drawing. This may be obtained from the ASME Data Report.
string	Manufacturer	("Manufacturer", dtString, Maximum Length 50)
		From a lookup list, select the name of the manufacturer. This can be obtained from the ASME Data Report or equipment stamping.
string	ManufSerialNo	("Mfr Serial No", dtString, Maximum Length 50)
		The manufacturer serial number. This may be obtained from the ASME Data Report or the equipment nameplate stamping.
string	ModelNumber	("Model Number", dtString, Maximum Length 35)
		The model number for the equipment.
string	NationalBoardNum	("NB", dtString, Maximum Length 15)
		The National Board registration number. This may be obtained from the ASME Data Report and usually for boilers built outside of Canada.
string	OutageExtent	("Outage Extent", dtString, Maximum Length 20)
		From a lookup list, select the extent of outage for a shutdown. Identifies if total plant, unit, or equipment outage is required for inspection
string	PIDNumber	("PID Number", dtString, Maximum Length 250)
		The Piping and Instrumentation Drawing number for the equipment.
decimal?	RegistrationAmount	("Registration Amount", dtDouble, Unit-Based Field)
		The amount paid for the registration to a regulatory board.
string	RegistrationFee	("Registration Fee", dtString, Maximum Length 20)

		From a lookup list, select the type of registration fee to a regulatory board. Choices are: Definition Initial – indicates that the registration fee is a one-time initial charge for the asset's registration Annual – indicates that the registration fee is an annual recurring charge for the asset's registration None – indicates that there is no fee for the asset's registration
string	RegistrationNo	("Registration Number", dtString, Maximum Length 30)
		The provincial, state or insurance company registration number.
string	ShapeType	("ASME Type", dtString, Maximum Length 15)
		From a lookup list, select the ASME designation of the equipment. This may be obtained from the ASME Data Report (watertube, firetube, etc.).
string	Status	("Status", dtString, Maximum Length 25)
		From a lookup list, select the status of the equipment. Examples: In Service, Out of Service, Scrapped, etc. Limitation: For assets associated with equipment types derived from the Pipeline Segment, Pipeline Custom or Wellhead equipment types, the call must be specifically formatted to include the Status Comments. Reference the Call Examples > Change Pipeline Status example for the required format.
string	TrainNum	("Train / Unit", dtString, Read-Only Field, Maximum Length 35)
		The train or unit of the plant where the equipment is located. Note: Validates against the Train/Units configured for the identified Plant.
int?	YearBuilt	("Year Built", dtInteger)
		The year of the equipment construction. This can be obtained from the ASME Data Report or the equipment nameplate.
int?	YearInstalled	("Year Installed", dtInteger)
		The year the equipment was installed in that specific location.

Foreign Reference		
Table Name	**Call Value**	**Definition**
BoilerStatic	rfBoilerStatic	Boiler Static Data - all equipment using the supplied Boiler equipment type
BundleStatic	rfBundleStatic	Bundle Static Data Static Data - all equipment using the supplied Bundle equipment type
CPStat	rfCPStat	CP Static Data - all equipment using the supplied CP System, CP Rectifier, CP Custom equipment types
CTStatic	rfCTStatic	Cooling Tower Static Data - all equipment using the supplied Cooling Tower equipment type
DesMat	rfDesMat	Static Data > Design and Materials
EquipDL	rfEquipDL	Equipment DataLogger ID - assign a unique identifier to provide a standard naming convention when tubes are surveyed using the VisLogger (datalogging) application. **Limitation**: Available when the Tube Trending module is active and equipment derived from the supplied Boiler and Furnace equipment types.

ExStatic	rfExStatic	Exchanger Static Data - all equipment using the supplied Exchanger equipment type
FilterStat	rfFilterStat	Filter Static Data - all equipment using the supplied Filter equipment type
FrStatic	rfFrStatic	Furnace Static Data - all equipment using the supplied Furnace equipment type
FTBoilStat	rfFTBoilStat	Firetube Boiler Static Data - all equipment using the supplied Firetube Boiler equipment type
GeoLoc	rfGeoloc	Static Data > Geolocation Coordinates - both original GPS and Linear Note: Linear Coordinates are Only updatable when the asset belongs to a Pipeline Plant.
InjPntStat	rfInjPntStat	Injection Point Static Data - all equipment using the supplied Injection Point equipment type
MiscStat	rfMiscStat	Misc Static Data - all equipment using the supplied Misc Custom and Pipeline Custom equipment types
PipeStat	rfPipeStat	Pipe Static Data - all equipment using the supplied Pipe equipment type
PL_LocationRef	rfPL_LocationRef	UTM References - all equipment using the supplied Pipeline Segment, Wellhead and Pipeline Custom equipment types;
PL_SegmentStat	rfPL_SegmentStat	Pipeline Segment Static Data - all equipment using the supplied Pipeline Segment equipment type
PL_WellheadStat	rfPL_WellheadStat	Wellhead Static Data - all equipment using the supplied Wellhead equipment type
PltExchStat	rfPltExchStat	Plate Exchanger Static Data - all equipment using the supplied Plate Exchanger equipment type
PSVStat	rfPSVStat	PSV Static Data - all equipment using the supplied PSV and PSV Location equipment types
RE_Couplings	rfRE_Couplings	RE Couplings - all equipment using the supplied GP Stm Turbine, Centrifugal Pump, and Motor equipment types
RE_GPTurbStat	rfRE_GPTurbStat	GP Steam Turbine Static Data - all equipment using the supplied GP Stm Turbine equipment type
RE_MotorStat	rfRE_MotorStat	Motor Static Data - all equipment using the supplied Motor equipment type
RE_PumpStat	rfRE_PumpStat	Centrifugal Pump Static Data - all equipment using the supplied Centrifugal Pump equipment type
Schedule	rfSchedule	Scheduling - equipment level
StackStat	rfStackStat	Stack Static Data - all equipment using the supplied Stack equipment type
ST_CraneStat	rfST_CraneStat	Structures Crane Static Data – all equipment using the supplied ST Crane equipment type
ST_FixedUWStat	rfST_FixedUWStat	Structures Fixed U/W Static Data – all equipment using the supplied ST Fixed Underwater equipment type
ST_HelipadStat	rfST_HelipadStat	Structures Helipad Static Data – all equipment using the supplied ST Helipad equipment type
ST_TopsideStat	rfST_TopsideStat	Structures Topside Static Data – all equipment using the supplied ST Topside equipment type
TankStatic	rfTankStatic	Tank Static Data - all equipment using the supplied Tank equipment type
ValveStat	rfValveStat	Valve Static Data - all equipment using the supplied Valve equipment type

Vessel	rfVessel	Vessel Static Data - all equipment using the supplied Vessel equipment type
CoatLin	lstCoatLin	Coatings - list or table of coatings / linings / cladding for each equipment **Limitation**: Not available for equipment derived from the supported Pipe and Pipeline Segment equipment types; use lstGL_CoatLin_V.
Elevations	lstElevations	Elevations - list or table of elevations for each equipment **Limitation**: Available when the Tube Trending module is active and equipment derived from the supplied Boiler and Furnace equipment types.
EquipRegAuth	lstEquipRegAuth	Regulatory Authorities - list or table of authorities for each equipment based on the supported equipment types
FuelTypes	lstFuelTypes	Fuel Types - list or table of types of fuel for each equipment based on the supported equipment types
GL_CoatLin_V	lstGL_CoatLin_V	Coatings w/Coordinates - list or table of coatings / linings / cladding with geolocation coordinates for each equipment **Limitation**: Available for equipment derived from the supported Pipe and Pipeline Segment equipment types.
NotesStat	lstNotesStat	Notes - list or table of notes for each equipment
Nozzle	lstNozzle	Nozzles - list or table of nozzles for each equipment based on the supported equipment types
RE_Bearings	lstRE_Bearings	Bearings - list or table of bearings for each equipment **Limitation**: Available for equipment derived from the supported Centrifugal Pump and Motor equipment types.
SectionDesign	lstSectionDesign	Section Design - list or table of sections for each equipment based on the supported equipment types
TrayPack	lstTrayPack	Internals - list or table of internals for each equipment based on the supported equipment types

DesMat (rfDesMat)

Design and Materials: the centralized design and material information for all assets within a site

Limitation: The fields listed below are not used by all types of equipment, for a list of the fields that apply to each individual type of equipment use the Field-Equipment Type cross reference below.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
decimal?	Area	("Area", dtDouble, Unit-Based Field)
		The area or heating surface of the equipment. This can be obtained from the ASME Data Report.
decimal?	Capacity	("Volume/Capacity", dtDouble, Unit-Based Field)
		The volume or capacity of the equipment, in appropriate units. This can be obtained from the ASME Data Report, the manufacturer drawings or engineering specifications.
string	Code	("Code", dtString, Maximum Length 10)
		From a lookup list, select the applicable Code of construction. This may be obtained from the equipment Data Report, engineering specifications or drawings.
string	CodeAddenda	("Addenda", dtString, Maximum Length 10)

		The year of the applicable addenda for the Code of construction. This can be obtained from the equipment Data Report, engineering specifications or drawings.
string	CodeDivision	("Division", dtString, Maximum Length 10)
		From a lookup list, select the Division of the Section of the Code of construction. This may be obtained from the equipment Data Report, engineering specifications or drawings.
string	CodeSection	("Section", dtString, Maximum Length 10)
		From a lookup list, select the Section of the Code of construction. This may be obtained from the equipment Data Report, engineering specifications or drawings.
string	CodeStamp	("Stamp", dtString, Maximum Length 10)
		From a lookup list, select the applicable ASME Code stamp of construction. This may be obtained from the equipment Data Report, engineering specifications or drawings.
int?	CodeYear	("Year", dtInteger)
		The year of edition of the applicable Code of construction. This can be obtained from the equipment Data Report, engineering specifications or drawings.
decimal?	Depth	("Depth", dtDouble, Unit-Based Field)
		The overall depth of the equipment, in appropriate units. This can be obtained from the ASME Data Report.
int?	DesignLife	("Design Life", dtInteger, Unit-Based Field)
		The number of years the equipment is designed for.
decimal?	DesignPress	("Design Press", dtDouble, Unit-Based Field)
		The design pressure of the equipment. This can be obtained in the design specifications or drawings.
decimal?	DesignTemp	("Design Temp", dtDouble, Unit-Based Field)
		The design temperature of the equipment. This can be obtained in the design specifications or drawings
decimal?	Diameter	("Diameter", dtDouble, Unit-Based Field)
		The diameter of the equipment; this is typically the inside diameter. This can be obtained in the ASME Data Report.
decimal?	FilledWeight	("Filled Weight", dtDouble, Unit-Based Field)
		The weight of the equipment when filled with water. This would be obtained from the ASME Data Report, the manufacturer drawings or engineering specifications.
bool?	FullVacuum	("Full Vacuum", dtInteger)
		Indicate whether the equipment was designed for full vacuum. This information can be found in the equipment data report, design specifications or drawings. Choices are: 0, N, No, 1, Y, Yes.
bool?	Handhole	("Handhole", dtInteger)
		Indicate whether if there are handholes installed in the equipment. This is obtained from the ASME Data Report or the manufacturer construction drawings. Choices are: 0, N, No, 1, Y, Yes.

string	HowAttached	("How Attached", dtString, Maximum Length 20) From a lookup list, select how the support(s) are attached to the equipment. This may be obtained from the ASME Data Report.
string	HowLengthTaken	("How Length Taken", dtString, Maximum Length 15) From a lookup list, select how the length of the equipment was taken.
decimal?	ItemWeight	("Empty Weight", dtDouble, Unit-Based Field) The weight of the equipment when empty. This can be obtained from the ASME Data Report, the manufacturer drawings or engineering specifications.
decimal?	LengthHeight	("Length/Height", dtDouble, Unit-Based Field) The overall length or height of the equipment, in appropriate units. This can be obtained from the ASME Data Report.
bool?	Manway	("Manway", dtInteger) Indicate whether there are manways installed in the equipment. This can be obtained from the ASME Data Report or the manufacturer construction drawings. Choices are: 0, N, No, 1, Y, Yes.
decimal?	MAWP	("MAWP", dtDouble, Unit-Based Field) The Maximum Allowable Working Pressure of the equipment. This can be obtained from the ASME Data Report or the equipment nameplate.
decimal?	MDMT	("MDMT", dtDouble, Unit-Based Field) The Minimum Design Metal Temperature. This can be obtained from the ASME Data Report or the equipment nameplate.
decimal?	MDMTPress	("MDMT Press", dtDouble, Unit-Based Field) The maximum pressure allowed at the Minimum Design Metal Temperature. This can be obtained from the ASME Data Report or the equipment nameplate.
decimal?	MDWT	("MDWT", dtDouble, Unit-Based Field) The Minimum Design Working Temperature. This can be obtained from the ASME Data Report or the equipment nameplate.
decimal?	NominalThickness	("Nominal", dtDouble, Unit-Based Field) The actual wall thickness of the equipment. This can be obtained from the ASME Data Report.
int?	NumOfPasses	("Num of Passes", dtInteger) The number of passes in the equipment.
decimal?	OperatingLimit	("Operating Limit", dtDouble, Unit-Based Field) The operating limit of the equipment; in hours. This is used in conjunction with the Operating Time in the Inspection Report to identify an operating threshold.
decimal?	OperatingPressure	("Operating Press", dtDouble, Unit-Based Field) The normal operating pressure of the equipment. This can be obtained from Operations.
decimal?	OperatingTemp	("Operating Temp", dtDouble, Unit-Based Field) The normal operating temperature of the equipment. This can be obtained from Operations
decimal?	OpPressOut	("Operating Press Out", dtDouble, Unit-Based Field)

		The normal operating outlet pressure of the equipment. This can be obtained from Operations.
decimal?	OpTempOut	("Operating Temp Out", dtDouble, Unit-Based Field)
		Ente the normal operating outlet temperature of the equipment. This can be obtained from Operations.
decimal?	OrigTestPress	("Orig Test Press", dtDouble, Unit-Based Field)
		The original pressure test pressure during construction measured at the top of the equipment. This can be obtained from the ASME Data Report.
string	Other1	("Other 1", dtString, Maximum Length 50)
		The value as defined by the Visions administrator. A generic field to enter other specific information. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other2	("Other 2", dtString, Maximum Length 50)
		The value as defined by the Visions administrator. A generic field to enter other specific information. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other3	("Other 3", dtString, Maximum Length 50)
		The value as defined by the Visions administrator. A generic field to enter other specific information. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other4	("Other 4", dtString, Maximum Length 50)
		The value as defined by the Visions administrator. A generic field to enter other specific information. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	OtherAccess	("Other Access", dtString, Maximum Length 20)
		From a lookup list, select another type of access to the equipment.
string	PaintCode	("Paint Code", dtString, Maximum Length 15)
		From a lookup list, select the paint code for a piece of equipment.
bool?	ShellNonPress	("Shell Non-Press", dtInteger)
		Indicate whether the shell side of the equipment is considered non-pressure or not. This can be obtained from the data report or design specifications. Choices are: 0, N, No, 1, Y, Yes.
string	ShellNPReason	("Shell NP Reason", dtString, Maximum Length 25)
		From a lookup list, select the reason why (vented to atmosphere, etc.) the shell-side is considered non-pressure. This may be obtained from the data report or design specifications.
bool?	SourService	("Sour Service", dtInteger)
		Indicate whether the equipment is in sour service. This may be obtained from the design specifications or drawings. Choices are: 0, N, No, 1, Y, Yes.
string	Support	("Type of Support", dtString, Maximum Length 20)
		From a lookup list, select the basic type of support for the equipment. This may be obtained from the ASME Data Report.

decimal?	TempMAWT	("MAWT", dtDouble, Unit-Based Field)
		The Maximum Allowable Working Temperature. This may be obtained from the ASME Data Report or the equipment nameplate.
string	TestType	("Orig Test Type", dtString, Maximum Length 15)
		From a lookup list, select the type of original construction pressure test. Examples: Hydrostatic, Pneumatic, Combination. This may be obtained from the ASME Data Report.
bool?	TubeNonPress	("Tube Non-Press", dtInteger)
		Indicate whether the tube side of the equipment is considered non-pressure. This may be obtained from the Data Report or design specifications. Choices are: 0, N, No, 1, Y, Yes.
string	TubeNPReason	("Tube NP Reason", dtString, Maximum Length 25)
		From a lookup list, select the reason why (vented to atmosphere, etc.) the tube-side is considered non-pressure. This may be obtained from the data report or design specifications.
string	WhereAttached	("Where Attached", dtString, Maximum Length 20)
		From a lookup list, select where the supports are located on the equipment. This may be obtained from the ASME Data Report.
decimal?	Width	("Width", dtDouble, Unit-Based Field)
		The overall width of the equipment, in appropriate units. This may be obtained from the ASME Data Report.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

Field-Equipment Type Cross Reference

Field	Boiler	Bundle	Centrifugal Pump	Circuit	Cooling Tower	CP Custom	CP Rectifier	CP System	Exchanger	Filter	Firetube Boiler	Furnace	GP Steam Turbine	Injection Point	Major	Misc Custom	Motor	Pipe	Plate Exchanger	PL Custom	PL Segment	PSV	PSV Location	Stack	ST Topside	ST Fixed U/W	ST Helipad	ST Crane	Tank	Valve	Vessel	Wellhead
Area									*		*								*													
Capacity			*						*	*	*								*										*		*	
Code	*	*	*	*	*	*	*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
CodeAddenda	*	*	*	*	*	*	*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
CodeDivision	*	*	*	*	*	*	*		*	*	*	*	*	*	*	*	*	*	*	*		*	*	*	*	*	*	*	*	*	*	*
CodeSection	*	*	*	*	*	*	*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
CodeStamp	*	*	*	*	*	*	*		*	*	*	*	*	*	*	*	*	*	*	*		*	*	*	*	*	*	*		*	*	*
CodeYear	*	*	*	*	*	*	*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		*	*	*
Depth						*				*		*				*			*													
DesignLife	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*			*	*	*	*	*	*	*	*	*
DesignPress	*	*		*					*	*	*	*			*	*													*		*	
DesignTemp	*	*		*					*	*	*	*			*	*													*		*	
Diameter		*				*				*		*				*													*		*	*
FilledWeight	*								*	*									*										*		*	
FullVacuum									*	*																					*	
Handhole	*								*	*	*	*				*			*												*	
HowAttached	*								*	*	*	*				*			*						*	*	*	*			*	
HowLengthTaken																																
ItemWeight	*								*	*	*								*										*		*	
LengthHeight		*			*	*			*	*		*				*			*					*					*		*	*
Manway	*								*	*	*	*				*													*		*	
MAWP		*	*	*					*	*		*			*	*								*							*	
MDMT		*		*					*	*		*			*																*	
MDMTPress		*		*					*	*		*			*																*	
MDWT																																
NominalThickness					*																*											
NumOfPasses									*							*																
OperatingLimit	*	*		*	*				*	*	*	*			*	*		*	*		*			*					*	*	*	*
OperatingPressure		*		*					*	*	*	*			*	*				*				*					*		*	
OperatingTemp		*		*	*				*	*	*	*			*	*				*				*					*		*	
OpPressOut												*				*																
OpTempOut					*				*			*																				
OrigTestPress	*	*							*	*	*	*																			*	
Other1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Other2	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Other3	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Other4	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
OtherAccess	*								*	*	*	*				*			*						*	*	*	*	*		*	
PaintCode	*	*		*	*				*	*	*	*			*	*		*	*		*			*	*	*	*	*	*		*	*
ShellNonPress									*	*						*															*	
ShellNPReason									*	*						*															*	
SourService									*	*						*		*											*		*	
Support	*								*	*	*	*				*			*						*	*	*	*			*	
TempMAWT		*	*	*					*	*		*			*	*								*							*	
TestType	*	*							*	*	*	*																			*	
TubeNonPress									*							*		*														
TubeNPReason									*							*		*														
WhereAttached	*								*	*	*	*				*			*						*	*	*	*			*	
Width						*				*		*				*			*													

Schedule (rfSchedule)

Schedule: centralized scheduling information for all assets within a site

Limitation: Some fields are dependent upon the RBI module being active.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
int?	CARCID	("CA RC ID", dtInteger, Read-Only Field)

		From a lookup list, select the risk assessment identification number for the criticality classification of an asset. **Note**: If the RBI module is active and a criticality risk assessment is not applied to the asset, the value will be validated as an active and completed "Criticality" risk configuration type. **Limitations**: If the RBI module is not active, a message will be logged within a call identifying that the CA RC ID cannot be updated. If a risk assessment has been applied to an asset within the Visions Client, a message will be logged within a call identifying that the CA RC ID cannot be updated.
int?	CARISKCODE	("Criticality IRC", dtInteger)
		From a lookup list, select the Inspection (Risk) Code for the criticality classification of an asset based on the criticality risk configuration (CARCID). **Note**: If the RBI module is active, the CA RC ID must be populated to set the Criticality IRC. **Limitation**: If a criticality risk assessment has been applied to an asset within the Visions Client, a message will be logged within a call identifying that the CA Risk Code cannot be updated.
int?	ExtInterval	("External Interval", dtInteger, Unit-Based Field)
		The external inspection interval for a piece of equipment.
DateTime?	InServiceDate	("In Service Date", dtDate)
		The date the equipment was put in service. **Limitation**: Only applies to non-PSV equipment (equipment types not derived from PSV and PSV Location types).
int?	InspRiskCode	("IRC", dtInteger)
		From a lookup list, select the Inspection (Risk) Code for equipment based on the selected risk configuration (RCID). Note: If the RBI module is active, the RC ID must be populated to set the Inspection Risk Code. **Limitation**: If a risk assessment has been applied to an asset within the Visions Client, a message will be logged within a call identifying that the Inspection Risk Code cannot be updated.
DateTime?	InstallDate	("Install Date", dtDate)
		The date the equipment was put in service; either initially or after an outage. **Limitation**: Only applies to equipment with equipment types derived from PSV and PSV Location types.
DateTime?	NextExtDueDate	("Next Ext Due Date", dtDate)
		Enter a date for the next external inspection date for a piece of equipment. **Limitation**: When setting this value, the NextExtReason and OverwriteJustification values are also required.
string	NextExtReason	("External Overwrite Reason", dtString, Maximum Length 25)
		From a lookup list, select the reason for changing the default Next External Inspection Due date. **Limitation**: Only set and is required when the NextExtDueDate is changed.
string	NextFullInspReason	("Overwrite Reason", dtString, Maximum Length 25)

		From a lookup list, select the reason for changing the default Next Full Inspection Due date. **Limitation**: Only set and is required when the NextInspFullDue is changed.
DateTime?	NextInspFullDue	("Next Full Insp Due", dtDate) Select a date for the next full inspection for a piece of equipment. It initially defaults to the Next Full Inspection (Calculated) date. **Limitation**: When setting this value, the NextFullInspReason and OverwriteJustification values are also required.
DateTime?	OutageDate	("Outage Date", dtDate) The outage date to denote when the equipment will be shut down for upcoming inspections.
string	OverwriteJustification	("Justification", dtString, Maximum Length 4000) The justification for the overwriting the scheduled due date.
string	PMNo_Ext	("External PM No", dtString, Maximum Length 25) From a lookup list, select the ERP PM (preventative maintenance) number for an external inspection, related to the Next External date. **Limitation**: Only updatable when the ERP-EAM module is active.
string	PMNo_Full	("Full PM No", dtString, Maximum Length 25) The ERP PM (preventative maintenance) number for a full or internal inspection, related to the Next Full Insp dates. **Limitation**: Only updatable when the ERP-EAM module is active.
int?	RAID	("RC ID", dtInteger) From a lookup list, select the risk assessment identification number when either a risk assessment is applied to the equipment or an Inspection Risk Code is selected. **Note**: If the RBI module is active, the equipment type for the asset is derived from PSV supplied types, and PSVs are configured to use their own RBI configuration, the value will be validated as an active and completed "PSV" risk configuration type. If the RBI module is active, and the equipment type for the asset is derived from non-PSV supplied types, the value will be validated as an active and completed "Standard" risk configuration type. **Limitation**: Only active, completed risk configurations may be associated within Scheduling. If a risk assessment has been applied to the asset within the Visions Client, a message will be logged within a call identifying that the RC ID cannot be updated.
DateTime?	ReassessDate	("Reassessment Date", dtDate) The date to perform the RBI reassessment for the equipment.
int?	RestrictedInterval	("Restricted Interval", dtInteger, Unit-Based Field) The inspection interval for a piece of equipment. It is defaulted when an Inspection Risk Code is selected or Risk Assessment is applied but can be overwritten. Note: A whole number greater than or equal to 1 represents the interval in months
decimal?	RLFactor	("RL Factor", dtDouble) The remaining life factor for a piece of equipment. It is defaulted when an Inspection Risk Code is selected or Risk Assessment is applied but can be overwritten. **Note**: A fractional number between 0 and 1; allowing 2 decimal places.

int?	ScheduleFlag	("Driven by Minor", dtInteger) Indicate whether the scheduling for Major equipment is driven by the minor equipment. **Limitation**: Only updatable when the equipment type for the owning asset is derived from the Major supplied type.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

GeoLoc (rfLocGUID)

Geolocation Coordinates: the centralized original GPS and linear coordinates for equipment, work, and trending.

Limitation: The fields listed below are based on the Plant to which the equipment belongs. If the plant is flagged as a Pipeline Plant, then the Linear coordinates are available; otherwise only the original GPS coordinates are available.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
bool?	CenterlineFlag	("Use in Centreline", dtInteger) Indicate whether the location coordinates are to be used as part of the dynamic centerline. Acceptable values are: 0, N, No, 1, Y, Yes; defaults to N. Only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
string	Desc_End	("Description - End", dtString, Maximum Length 100) The description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100) The description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the ending milepost distance (chainage). The lookup list is extracted from the PL_SegMarkers Reference Markers. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). The lookup list is extracted from the PL_SegMarkers Reference Markers. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field) The ending chainage or linear location.

		Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field) The starting chainage or linear location. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger) From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage. Choices are: Left, Right. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger) From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage. Choices are: Left, Right. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field) The original ending GPS Longitude (X) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field) The original starting GPS Longitude (X) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Y_EndOrig	("Latitude - End", dtDouble, Unit-Based Field) The original ending GPS Latitude (Y) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field) The original starting GPS Latitude (Y) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field) The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - End", dtDouble, Unit-Based Field) The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment
GL_CoatLin_V	lstGL_CoatLin_V
GL_LineDetail_V	lstGL_LineDetail_V
GL_PLBends_V	lstGL_PLBends_V
GL_PLControls_V	lstGL_PLControls_V
GL_PLSegCrossings_V	lstGL_PLSegCrossings_V
GL_PLSegmentLD_V	lstGL_PLSegmentLD_V
GL_PLSoilInfo_V	lstGL_PLSoilInfo_V
GL_PLTopography_V	lstGL_PLTopography_V
GL_PLWaterDepth_V	lstGL_PLWaterDepth_V

BoilerStatic (rfBoilerStatic)

Boiler Static: the static information associated with equipment types derived from the Boiler supplied type

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	DraftType	("Draft Type", dtString, Maximum Length 20)
		From a lookup list, select the type of draft for the boiler. Examples: Natural, Forced, Induced, Balanced.
bool?	FieldErected	("Field Erected", dtInteger)
		Indicate whether the equipment was field erected or not. Choices are: 0, N, No, 1, Y, Yes.
int?	NumOfBurners	("Num of Burners", dtInteger)
		The number of burners for a boiler.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter boiler equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all boiler static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter boiler equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all boiler static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
bool?	ShopFabricated	("Shop Fabricated", dtInteger)
		Indicate whether the boiler was shop fabricated or not. Choices are: 0, N, No, 1, Y, Yes.
decimal?	StmCapacity	("Stm Capacity", dtDouble, Unit-Based Field)
		The manufacturer certified steaming capacity. This can be obtained from the ASME Data Report or the boiler nameplate.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment
BoOpening	lstBOOpening
HeatSurface	lstHeatSurface

BundleStatic (rfBundleStatic)

Bundle Static: the static information associated with equipment types derived from the Bundle supplied type

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	BundleType	("Bundle Model/ID", dtString, Maximum Length 20)
		From a lookup list, select the bundle model or ID number. For removable bundles, the intent is to indicate the exchangers each bundle fits into. This may be obtained from the manufacturer construction drawings or engineering specs.
decimal?	BundleWeight	("Bundle Weight", dtDouble, Unit-Based Field)
		The weight of the bundle when empty. This can be obtained from the ASME Data Report, the manufacturer drawings or engineering specifications.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter bundle equipment type-specific data. These fields can be re-named in the Administration module, the name will appear on all bundle static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter bundle equipment type-specific data. These fields can be re-named in the Administration module, the name will appear on all bundle static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	TubeType	("Tube Type", dtString, Maximum Length 20)
		From a lookup list, select the type of tubing. Examples: straight tube, U tube, fin-fan, etc. This may be from the ASME Data Report or the manufacturer construction drawings.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

CPStat (rfCPStat)

Cathodic Protection Static: the static information associated with equipment types derived from the CP Custom, CP Rectifier, and CP System supplied types.

Limitation: The fields listed below are not used by all CP types of equipment, for a list of the fields that apply to each individual type of equipment it is recommended that the Excel Wizard Static Data templates be used as a reference.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	AnodeClass	("Class", dtString, Maximum Length 20) From a lookup list, select the class of anode.
decimal?	AnodeDistToStruct	("Distance to Structure", dtDouble, Unit-Based Field) The distance from the rectifier to the structure. If more than one structure is protected, The distance for the closest structure.
string	AnodeInstallType	("Installation Type", dtString, Maximum Length 20) From a lookup list, select the type of installation of an anode.
decimal?	AnodeMoisture	("Moisture", dtDouble, Unit-Based Field) The moisture percentage in the soil for the anodes or ground bed.
string	BackfillMaterial	("Backfill Material", dtString, Maximum Length 25) From a lookup list, select the type of material used in backfill.
decimal?	BackfillResistivity	("Resistivity", dtDouble, Unit-Based Field) The amount of resistance provided by the backfill.
string	BackfillType	("Backfill Type", dtString, Maximum Length 20) From a lookup list, select the type of backfill for a ground-bed or anodes; such as: Chemical, Carbonaceous.
decimal?	CritSizeDiameter	("Crit Size Diameter", dtDouble, Unit-Based Field) The diameter of the equipment.
decimal?	CritSizeLength	("Crit Size Length", dtDouble, Unit-Based Field) The length of the equipment.
decimal?	CritSizeWidth	("Crit Size Width", dtDouble, Unit-Based Field) The width of the equipment.
bool?	Groundbed	("Groundbed", dtInteger) Indicate whether the anodes are stored in a ground bed. Choices are: 0, N, No, 1, Y, Yes.
string	GroundbedType	("Groundbed Type", dtString, Maximum Length 20) From a lookup list, select the type of groundbed; such as Deep, Semi-deep, Horizontal.
string	Material	("Material", dtString, Maximum Length 35) From a lookup list, select the type of material for the equipment.
int?	NoOfAnodes	("Num of Anodes", dtInteger) The quantity of anodes connected to the rectifier.
string	Other5	("Other 5", dtString, Maximum Length 50) Enter miscellaneous equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all miscellaneous static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)

		Enter miscellaneous equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all miscellaneous static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
decimal?	RectACVolts	("AC Volts", dtDouble, Unit-Based Field) The AC voltage for a rectifier.
string	RectAdjustType	("Adjustment Type", dtString, Maximum Length 20) From a lookup list, select the type of adjustment for a rectifier; such as Tap.
decimal?	RectFrequency	("Frequency", dtDouble, Unit-Based Field) The supply frequency for a rectifier.
decimal?	RectInputCurrent	("Input Current", dtDouble, Unit-Based Field) The maximum input current for a rectifier.
decimal?	RectInputVoltage	("Input Voltage", dtDouble, Unit-Based Field) The rated input voltage for the rectifier.
decimal?	RectOutputCurrent	("Output Current", dtDouble, Unit-Based Field) The maximum output current for the rectifier.
decimal?	RectOutputVoltage	("Output Voltage", dtDouble, Unit-Based Field) The rated output voltage for the rectifier.
string	RectPhase	("Phase", dtString, Maximum Length 20) The number of phases in a rectifier; such as single-phase or three-phase.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

CTStatic (rfCTStatic)

Cooling Tower Static: the static information associated with equipment types derived from the Cooling Tower supplied type.

DataType	FieldCallValue	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
decimal?	BasinDepth	("Basin Depth", dtDouble, Unit-Based Field) The depth of the basin for the cooling tower.
decimal?	BasinLength	("Basin Length", dtDouble, Unit-Based Field) The length of the basin for the cooling tower.
decimal?	BasinWidth	("Basin Width", dtDouble, Unit-Based Field) The width of the basin for the cooling tower.
string	DraftType	("Draft Type", dtString, Maximum Length 20) From a lookup list, select the type of draft for the equipment. Examples: Natural, Forced, Induced, Balanced.
bool?	FieldErected	("Field Erected", dtInteger)

		Indicate whether the equipment was field erected or not. Choices are: 0, N, No, 1, Y, Yes.
string	Material	("Material", dtString, Maximum Length 15)
		From a lookup list, select the type of material for the cooling tower.
int?	NumberOfCells	("Num of Cells", dtInteger)
		The number of cells in the cooling tower.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter cooling tower equipment type-specific data. These fields can be re-named in the Administration module, the name will appear on all cooling tower static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter cooling tower equipment type-specific data. These fields can be re-named in the Administration module, the name will appear on all cooling tower static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
bool?	ShopFabricated	("Shop Fabricated", dtInteger)
		Indicate whether the equipment was shop fabricated or not. Choices are: 0, N, No, 1, Y, Yes.
string	TowerType	("Tower Type", dtString, Maximum Length 15)
		From a lookup list, select the type of tower for the equipment. Examples: Wet, Dry, Combination.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment
CTNozzles	lstCTNozzles

ExStatic (rfExStatic)

Exchanger Static: the static information associated with equipment types derived from the Exchanger supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	BundleType	("Bundle Model/ID", dtString, Maximum Length 20)
		From a lookup list, select the bundle model or ID number. For removable bundles, the intent is to indicate the exchangers each bundle fits into. This may be obtained from the manufacturer construction drawings or engineering specs.
decimal?	DesignPress	("Tube Design Press", dtDouble, Unit-Based Field)
		The calculated tubeside design pressure of the exchanger based on the nominal thicknesses. These values are obtained from the design specifications.
decimal?	DesignTemp	("Tube Design Temp", dtDouble, Unit-Based Field)
		The calculated tubeside design temperature of the exchanger based on the nominal thicknesses. These values are obtained from the design specifications.

int?	FixedRemove	("Bundle Type", dtInteger) From a system lookup list, select the type of bundle for the exchanger. Choices are: Fixed, Removable.
bool?	FullVacuum	("Full Vacuum", dtInteger) Indicate whether the exchanger is full vacuum. Choices are: 0, N, No, 1, Y, Yes.
decimal?	MAWP	("Tube MAWP", dtDouble, Unit-Based Field) The Maximum Allowable Working Pressure for the tubeside of the exchanger. This can be obtained from the ASME Data Report or the equipment nameplate.
decimal?	MAWT	("Tube MAWT", dtDouble, Unit-Based Field) The Maximum Allowable Working Temperature for the tubeside of the exchanger. This can be obtained from the ASME Data Report or the exchanger nameplate.
decimal?	MDMT	("Tube MDMT", dtDouble, Unit-Based Field) The Minimum Design Metal Temperature for the tubeside of the exchanger. This can be obtained from the ASME Data Report or the exchanger nameplate.
decimal?	MDMTPress	("Tube MDMT Press", dtDouble, Unit-Based Field) The maximum pressure allowed at the Minimum Design Metal Temperature for the tubeside of the exchanger. This could be obtained from the ASME Data Report or the equipment nameplate.
int?	NumOfCourses	("Num of Courses", dtInteger) The number of shell courses in an exchanger.
int?	NumOfPasses	("Num of Passes", dtInteger) The number of passes in an exchanger.
decimal?	OperatingPressure	("Tube Operating Press", dtDouble, Unit-Based Field) The normal operating pressure for the tubeside of an exchanger. This can be obtained from Operations.
decimal?	OperatingTemp	("Tube Operating Temp", dtDouble, Unit-Based Field) The normal operating temperature for the tubeside of an exchanger. This can be obtained from Operations.
decimal?	OpTempOut	("Operating Temp Out", dtDouble, Unit-Based Field) The normal outlet operating temperature for the tubeside of an exchanger. This can be obtained from Operations.
decimal?	OrigTestPress	("Orig Test Press", dtDouble, Unit-Based Field) The original pressure test pressure during construction for the tubeside of an exchanger. These values can be obtained from the ASME Data Report.
string	OrigTestType	("Orig Test Type", dtString, Maximum Length 15) From a lookup list, select the type of original construction pressure test. These values can be obtained from the ASME Data Report and are either hydrostatic, pneumatic or combination.
string	Other5	("Other 5", dtString, Maximum Length 50) Enter exchanger equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all exchanger static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.

string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter exchanger equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all exchanger static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
bool?	SourService	("Sour Service", dtInteger)
		Indicate whether the tubeside of an exchanger is in sour service. Choices are: 0, N, No, 1, Y, Yes.
string	TemaClass	("TEMA Class", dtString, Maximum Length 1)
		From a lookup list, select the TEMA (Tubular Exchanger Manufacturers Association) classes. Examples: R, C or B.
string	TemaType	("TEMA Type", dtString, Maximum Length 15)
		From a lookup list, select the TEMA (Tubular Exchanger Manufacturers Association) heat exchanger type or configuration.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

FilterStat (rfFilterStat)

Filter Static: the static information associated with equipment types derived from the Filter supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	DoorType	("Door Type", dtString, Maximum Length 20)
		From a lookup list, select the type of door for a filter.
int?	FilterQuantity	("Filter Element Quantity", dtInteger)
		The quantity of the filter media in a filter. This value depends on the type of filter media required.
string	FilterType	("Filter Element/Media Type", dtString, Maximum Length 20)
		From a lookup list, select the type of filter element or media type for a filter.
int?	NumOfCourses	("Num of Courses", dtInteger)
		The number of shell courses in a filter.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter filter equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all filter static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter filter equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all filter static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.

bool	QuickOpenDoor	("Quick Opening Door", dtInteger, Required Field)
		Indicate whether access to the filter media is through a quick-opening door. Choices are: 0, N, No, 1, Y, Yes. **Note**: Defaults to "0" (No, N) unless otherwise specified.
string	SpecialService	("Special Service", dtString, Maximum Length 25)
		From a lookup list, select the type of special service for a filter. Examples: Direct Fired, Lethal, Low Temperature.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment
FilterMedia	lstFilterMedia

FrStatic (rfFrStatic)

Furnace Static: the static information associated with equipment types derived from the Furnace supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
decimal?	Depth	("Depth", dtDouble, Unit-Based Field)
		The depth of the furnace.
string	DraftType	("Draft Type", dtString, Maximum Length 20)
		From a lookup list, select the type of draft for the furnace. Examples: Natural, Forced, Induced, Balanced.
int?	NumOfBurners	("Num of Burners", dtInteger)
		The number of burners for the furnace.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter furnace equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all furnace static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter furnace equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all furnace static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

FTBoilStat (rfFTBoilStat)

Firetube Boiler Static: the static information associated with equipment types derived from the Firetube Boiler supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	DraftType	("Draft Type", dtString, Maximum Length 20) From a lookup list, select the type of draft for the firetube boiler (example: natural, forced, induced or balanced).
int?	NumOfBurners	("Num of Burners", dtInteger) The number of burners for the firetube boiler.
string	Other5	("Other 5", dtString, Maximum Length 50) Enter firetube boiler equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all firetube boiler static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50) Enter firetube boiler equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all firetube boiler static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment
FTStays	lstFTStays
TubeData	lstTubeData

InjPntStat (rfInjPntStat)

Injection Point Static: the static information associated with equipment types derived from the Injection Point supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
bool?	ChemReaction	("Chemical Reaction", dtInteger) Identify whether a chemical reaction occurs at the injection point. Choices are: 0, N, No, 1, Y, Yes.
bool?	Corrosion	("Corrosion Possible", dtInteger) Identify whether corrosion is possible at the injection point. Choices are: 0, N, No, 1, Y, Yes.
bool?	Erosion	("Erosion Possible", dtInteger) Identify whether erosion is possible at the injection point. Choices are: 0, N, No, 1, Y, Yes.
bool?	Exothermic	("Exothermic", dtInteger) Identify whether the injection point is exothermic. Choices are: 0, N, No, 1, Y, Yes.
string	FluidDispMeth	("Fluid Dispersion Method", dtString, Maximum Length 15) From a lookup list, select the method of fluid dispersion for an injection point.
bool?	GasEvol	("Gas Evolution", dtInteger)

		Identify whether gas evolution occurs at the injection point. Choices are: 0, N, No, 1, Y, Yes.
string	InjectionFrom	("Injection From", dtString, Maximum Length 15)
		From a lookup list, select where the injection occurs for an injection point. Example: Top, Side, Bottom, etc.
string	InjectOrient	("Injection Orientation", dtString, Maximum Length 15)
		From a lookup list, select the orientation of the injection. Example: Co-current, Counter-current, Cross-current, etc.
decimal?	ISDroplet	("Required Droplet Size", dtDouble)
		The required droplet size in the injecting stream for an injection point.
bool?	ISFiltering	("Filtering Required", dtInteger)
		Identify whether filtering is required in the injecting stream of an injection point. Choices are: 0, N, No, 1, Y, Yes.
string	ISFlowContReqmt	("Flow Control Requirement", dtString, Maximum Length 25)
		From a lookup list, select the flow control requirement for the injecting stream of an injection point.
string	ISFlowMeasReqmt	("Flow Measurement Requirement", dtString, Maximum Length 25)
		From a lookup list, select the flow measurement requirement for the injecting stream of an injection point.
decimal?	ISFluidFract	("Fluid Type Fractions", dtDouble)
		The fraction of fluid type for the injecting stream of an injection point.
decimal?	ISFluidSolids	("Fluid Type Solids Content", dtDouble)
		The solids content of the injecting stream for an injection point.
string	ISFluidType	("Fluid Type", dtString, Maximum Length 15)
		From a lookup list, select the type of fluid in the injecting stream for an injection point.
bool?	ISIsolation	("Positive Isolation Required", dtInteger)
		Identify whether positive isolation is required in the injecting stream for an injection point. Choices are: 0, N, No, 1, Y, Yes.
bool?	ISOnStmClean	("Onstream Cleaning Required", dtInteger)
		Identify whether onstream cleaning is required for an injection point. Choices are 0, N, No, 1, Y, Yes.
decimal?	ISRate	("Injection Rate", dtDouble, Unit-Based Field)
		The normal injection rate for an injection point.
decimal?	ISRateMax	("Injection Rate Maximum", dtDouble, Unit-Based Field)
		The maximum injection rate for an injection point.
decimal?	ISRateMin	("Injection Rate Minimum", dtDouble, Unit-Based Field)
		The minimum injection rate for an injection point.
decimal?	ISRSRate	("Receiving Stream Rate", dtDouble)
		The receiving stream percentage for an injection point.
decimal?	ISTemp	("Temp", dtDouble, Unit-Based Field)
		The temperature of the injecting stream for an injection point.
decimal?	ISTempMax	("Temp Maximum", dtDouble, Unit-Based Field)

		The maximum temperature of the injecting stream for an injection point.
decimal?	ISTempMin	("Temp Minimum", dtDouble, Unit-Based Field)
		The minimum temperature of the injecting stream for an injection point.
string	ISType	("Stream Type", dtString, Maximum Length 15)
		From a lookup list, select whether the injection point is continuous or batch injection.
decimal?	LiquidFlash	("Liquid Flash Percent", dtDouble)
		The liquid flash percentage for an injection point.
string	NearAllowObj	("Nearest Obstacle", dtString, Maximum Length 15)
		From a lookup list, select the type of obstacle that is allowed near an injection point. Example: T/W, Elbow, Orifice, etc.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter injection point equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all injection point static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter injection point equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all injection point static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
bool?	OtherPrecip	("Other Fluid Precipitation", dtInteger)
		Identify whether other fluid precipitate for an injection point. Choices are: 0, N, No, 1, Y, Yes.
bool?	PhysReaction	("Physical Reaction", dtInteger)
		Identify whether there is a physical reaction in an injection point. Choices are: 0, N, No, 1, Y, Yes.
decimal?	RSFlowRate	("RS Flow Rate", dtDouble, Unit-Based Field)
		The flow rate in the receiving stream for an injection point.
string	RSFlowRegime	("Flow Regime", dtString, Maximum Length 15)
		From a lookup list, select the type of regime for the flow of an injection point.
decimal?	RSFlowVeloc	("RS Flow Velocity", dtDouble, Unit-Based Field)
		The velocity of the flow for an injection point.
string	RSFluidType	("Fluid Type", dtString, Maximum Length 15)
		From a lookup list, select the state or type of the fluid in the receiving stream of an injection point. Example: Gas, Liquid, etc.
decimal?	RSMixStream	("Mixed Stream Temp", dtDouble, Unit-Based Field)
		The temperature of the mixed stream for an injection point.
decimal?	RSPress	("Press", dtDouble, Unit-Based Field)
		The pressure of the receiving stream for an injection point.
decimal?	RSTempIn	("Temp In", dtDouble, Unit-Based Field)
		The temperature into the receiving stream of an injection point.

bool?	WaterPrecip	("Water Precipitation", dtInteger) Identify whether water precipitation is present in an injection point. Choices are: 0, N, No, 1, Y, Yes.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

MiscStat (rfMiscStat)

Miscellaneous Custom Static: the static information associated with equipment types derived from the Misc Custom and Pipeline Custom supplied types.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
decimal?	MaxFlowRate	("Max Flow Rate", dtDouble, Unit-Based Field) The maximum flow rate of product/fluid through the equipment. This information is found in the design specifications.
string	Other5	("Other 5", dtString, Maximum Length 50) Enter miscellaneous equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all miscellaneous static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50) Enter miscellaneous equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all miscellaneous static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

PipeStat (rfPipeStat)

Pipe Static: the static information associated with equipment types derived from the Pipe supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
bool?	AirToGround	("Soil to Air", dtInteger) Indicate whether the piping is air to ground. Choices are: 0, N, No, 1, Y, Yes.
string	APIClass	("API Class", dtString, Maximum Length 10) From a lookup list, select the piping service class for the process piping. Classes are identified in API 570, section 6.2.
string	ASMECategory	("ASME Category", dtString, Maximum Length 15)

		From a lookup list, select the fluid service for the process piping. Typically identified in ASME B31.3, section 305.
string	ASMESubCategory	("ASME SubCategory", dtString, Maximum Length 15)
		From a lookup list, select the fluid service sub category to indicate if the piping meets the ASME definitions for High Pressure, Severe Cyclic, etc.
bool?	Deadleg	("Deadleg", dtInteger)
		Indicate whether the piping contains a deadleg. Choices are: 0, N, No, 1, Y, Yes.
bool?	InjectionPoint	("Injection Point", dtInteger)
		Indicate whether the piping contains an injection point. The injection point may be identified as a separate piece of equipment and connected through the Linked Equipment. Choices are: 0, N, No, 1, Y, Yes.
string	ISONumber	("ISO Number", dtString, Maximum Length 250)
		The isometric drawing number for the piping.
string	LineClass	("Line Class", dtString, Maximum Length 15)
		From a lookup list, select the line classification for the piping.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter piping equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all piping static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter piping equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all piping static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	PipeLocation	("Pipe Location", dtString, Maximum Length 200)
		A brief description of the location for the piping.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment
GL_LineDetail_V	lstGL_LineDetail_V

PL_SegmentStat (rfPL_SegmentStat)

Pipeline Segment Static: the static information associated with equipment types derived from the Pipeline Segment supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
bool?	BatchInhibPerformable	("Batch Treatment Performable", dtInteger)
		Identify whether batch corrosion inhibitor injection is performable for a pipeline segment. Choices are: 0, N, No, 1, Y, Yes.
string	BoatLandLocation	("Boat Landing Location", dtString, Maximum Length 35)
		From a lookup list, select the location of the boat landing for a pipeline segment.

bool?	Bumper	("Bumper", dtInteger)
		Identify whether the pipeline segment has a bumper. Choices are: 0, N, No, 1, Y, Yes.
bool?	CloseToBoatLand	("Close to Boat Landing", dtInteger)
		Identify whether the pipeline segment is close to a boat landing. Choices are: 0, N, No, 1, Y, Yes.
bool?	CloseToWorkArea	("Close to Work Area", dtInteger)
		Identify whether the pipeline segment is close to a work area. Choices are: 0, N, No, 1, Y, Yes.
DateTime?	CommissionDate	("Commission Date", dtDate)
		The date of commission for a pipeline segment.
decimal?	DesignTemp	("Design Temp", dtDouble, Unit-Based Field)
		The design temperature for a pipeline segment.
DateTime?	DetectionDate	("Detection Date", dtDate)
		If sulphur is present, The date of detection for a pipeline segment.
decimal?	EPZRadius	("EPZ Radius", dtDouble, Unit-Based Field)
		The emergency planning zone (EPZ) radius for a pipeline segment.
string	FluidType	("Type of Fluid", dtString, Maximum Length 25)
		From a lookup list, select the type of outside fluid that is added to a pipeline segment.
decimal?	HCA	("HCA", dtDouble, Unit-Based Field)
		The high consequence area for a pipeline segment.
decimal?	Hydrotest	("Hydrotest", dtDouble, Unit-Based Field)
		The hydrotest pressure for a pipeline segment.
bool?	ILIPerformable	("ILI Performable", dtInteger)
		Identify whether inline inspection is performable for a pipeline segment. Choices are: 0, N, No, 1, Y, Yes.
bool?	LateralPipeline	("Lateral Pipeline", dtInteger)
		Identify whether the pipeline segment is lateral. Choices are: 0, N, No, 1, Y, Yes.
decimal?	LicensedH2S	("Licensed H2S Content", dtDouble, Unit-Based Field)
		The licensed hydrogen sulphide (H2S) content for a pipeline segment.
decimal?	LicensedLength	("Licensed Length", dtDouble, Unit-Based Field)
		The licensed length for a pipeline segment.
string	LineTypeID	("Line Type ID", dtString, Maximum Length 15, FK (PL_LineTypes.LineTypeID))
		Select the line type for a pipeline segment. The list is built from the Pipeline Line Types lookup data.
string	LocationClass	("Location Class", dtString, Maximum Length 15)
		From a lookup list, select the location class for a pipeline segment.
decimal?	MAOP	("MOP", dtDouble, Unit-Based Field)
		The maximum allowable operating pressure for a pipeline segment.
decimal?	MOP	("MOP", dtDouble, Unit-Based Field)

		The maximum operating pressure for a pipeline segment.
bool?	OffshoreFlag	("Offshore", dtInteger)
		Identify whether the pipeline segment is offshore. Choices are: 0, N, No, 1, Y, Yes.
decimal?	OperPress	("Operating Press", dtDouble, Unit-Based Field)
		The operating pressure for the pipeline segment.
decimal?	OperTemp	("Operating Temp", dtDouble, Unit-Based Field)
		The operating temperature for the pipeline segment.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter pipeline segment equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all pipeline segment static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter pipeline segment equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all pipeline segment static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
bool?	OutsideFluids	("Outside Fluids", dtInteger)
		Identify whether outside fluids are added to a pipeline segment. Choices are: 0, N, No, 1, Y, Yes.
string	OutsideHowAdded	("Outside Fluids - How Added", dtString, Maximum Length 15)
		From a lookup list, identify how outside fluids are added to a pipeline segment.
bool?	PiggingPerformable	("Pigging Performable", dtInteger)
		Identify whether pigging is performable for the pipeline segment. Choices are: 0, N, No, 1, Y, Yes.
string	PrimaryContractor	("Primary Contractor", dtString, Maximum Length 50)
		The name of the primary contractor for the pipeline segment.
decimal?	RedEPZRad	("Reduced EPZ Radius", dtDouble, Unit-Based Field)
		The reduced EPZ (emergency planning zone) radius for the pipeline segment.
bool?	Regulated	("Regulated", dtInteger)
		Identify whether the pipeline segment is regulated. Choices are: 0, N, No, 1, Y, Yes.
DateTime?	RetireDate	("Retire Date", dtDate)
		The date of retirement for the pipeline segment.
bool?	Riser	("Riser", dtInteger)
		Identify whether the pipeline segment is on a riser. Choices are: 0, N, No, 1, Y, Yes.
bool?	ROWMarkers	("ROW Markers", dtInteger)
		Identify whether ROW (right-of-way) markers identify the pipeline segment. Choices are: 0, N, No, 1, Y, Yes.

string	ROWMarkerType	("ROW Marker Type", dtString, Maximum Length 15)
		From a lookup list, select the type of right-of-way marker for the pipeline segment.
int?	ROWPatrolFreq	("ROW Patrol Frequency", dtInteger, Unit-Based Field)
		The patrol frequency for the right-of-way marker for the pipeline segment.
string	ROWPatrolMethod	("ROW Patrol Method", dtString, Maximum Length 15)
		From a lookup list, select the patrol method for the right-of-way marker for the pipeline segment.
string	SetBackReqmnt	("Sour Natural Gas Level", dtString, Maximum Length 5)
		From a lookup list, select the sour natural gas level code for a pipeline segment.
bool?	SulfurPresent	("Sulfur Present", dtInteger)
		Identify whether sulphur is present in the pipeline segment. Choices are: 0, N, No, 1, Y, Yes.
decimal?	WeakDesPress	("Weakest Des Press", dtDouble)
		The weakest design pressure for the pipeline segment.
string	WorkAreaType	("Type of Work Area", dtString, Maximum Length 15)
		From a lookup list, select the type of work area for a pipeline segment.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment
GL_PLBends_V	lstGL_PLBends_V
GL_PLControls_V	lstGL_PLControls_V
GL_PLSegCrossings_V	lstGL_PLSegCrossings_V
GL_PLSoilInfo_V	lstGL_PLSoilInfo_V
GL_PLTopography_V	lstGL_PLTopography_V
GL_PLWaterDepth_V	lstGL_PLWaterDepth_V
PL_SegElevations	lstPL_SegElevations

PL_WellheadStat (rfPL_WellheadStat)

Wellhead Static: the static information associated with equipment types derived from the Wellhead supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	ArtificialLift	("Artificial Lift", dtString, Maximum Length 15)
		From a lookup list, select the type of artificial lift for a wellhead.
string	BottomHoleLSD	("Bottom Hole LSD/Location", dtString, Maximum Length 35)
		The bottom hole LSD (legal survey description) or location for a wellhead.
decimal?	CircStringDepth	("Circulation String Depth", dtDouble, Unit-Based Field)
		The circulation string depth for a wellhead.
DateTime?	CompletionDate	("Completion Date", dtDate)

		Select the completion date for a wellhead.
DateTime?	DetectionDate	("Detection Date", dtDate)
		If sulfur is present, select the detection date for a wellhead.
DateTime?	InitialProdDate	("Initial Production Date", dtDate)
		Select the initial production date for a wellhead.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter wellhead equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all wellhead static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter wellhead equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all wellhead static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
decimal?	ProdCaseID	("Casing ID", dtDouble, Unit-Based Field)
		The inner diameter for the production casing of a wellhead.
decimal?	ProdTubeOD	("Tubing OD", dtDouble, Unit-Based Field)
		The outer diameter for the production tubing of a wellhead.
string	ProductionZone	("Production Zone", dtString, Maximum Length 25)
		From a lookup list, select the production zone for a wellhead.
bool?	SulfurPresent	("Sulfur Present", dtInteger)
		Identify whether sulfur is present in a wellhead. Choices are: 0, N, No, 1, Y, Yes.
string	SurfaceLSD	("Surface LSD/Location", dtString, Maximum Length 35)
		The surface LSD (legal survey description) or location for a wellhead.
decimal?	TubingID	("Tubing ID", dtDouble, Unit-Based Field)
		The inner diameter of the production tubing for a wellhead.
decimal?	WellDepth	("Well Depth", dtDouble, Unit-Based Field)
		The depth of the well for a wellhead.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

PltExchStat (rfPltExchStat)

Plate Exchanger Static: the static information associated with equipment types derived from the Plate Exchanger supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter plate exchanger equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all plate exchanger static data forms.

		Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter plate exchanger equipment type-specific data. Since these fields can be renamed in the Administration module, the name will appear on all plate exchanger static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment
PltExchStreams	lstPltExchStreams

PsvStat (rfPsvStat)

PSV Static: the static information associated with equipment types derived from the PSV and PSV Location supplied types.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	AccOther1	("Acc Other 1", dtString, Maximum Length 25)
		Enter any additional accessories for a PSV.
string	AccOther2	("Acc Other 2", dtString, Maximum Length 25)
		Enter any additional accessories for a PSV.
string	AccOther3	("Acc Other 3", dtString, Maximum Length 25)
		Enter any additional accessories for a PSV.
string	AccOther4	("Acc Other 4", dtString, Maximum Length 25)
		Enter any additional accessories for a PSV.
decimal?	ActualCapacity	("Nameplate/Actual Capacity", dtDouble)
		The nameplate or actual capacity for a PSV.
string	ActualCapacityUnit	("Nameplate/Actual Capacity Unit", dtString, Maximum Length 15, UoM Factors Unit [Capacity])
		From a lookup list, select the unit of measure for the nameplate or actual capacity of a safety valve.
decimal?	AreaCalculated	("Area Calculated", dtDouble, Unit-Based Field)
		The calculated area for a PSV.
decimal?	AreaSelected	("Area Selected", dtDouble, Unit-Based Field)
		The selected area for a PSV.
string	BackFlowPreventer	("Backflow Preventer", dtString, Maximum Length 25)
		The type of backflow preventer for a PSV, if part of the accessories.
string	Bellows	("Bellows", dtString, Maximum Length 25)
		From a lookup list, select the type of bellows for a safety valve.
string	BlowDownSet	("Blowdown Set", dtString, Maximum Length 25)

		The blowdown percentage (the difference between the set and re-seat pressures) for a PSV.
string	BodyOther	("Body Other", dtString, Maximum Length 25)
		Enter any additional body information for a PSV.
string	BonnetStyle	("Bonnet Style", dtString, Maximum Length 25)
		From a lookup list, select the style of bonnet for a safety valve.
decimal?	BuiltUpBp	("Builtup BP", dtDouble, Unit-Based Field)
		The built-up back pressure (the outlet pressure caused by flow through the valve) for a PSV.
string	CarsealNumber	("Inlet Carseal No", dtString, Maximum Length 25)
		The inlet carseal number for a PSV.
bool?	CarsealOpen	("Inlet Carseal Locked Open", dtInteger)
		Identify if the inlet carseal is locked open for a PSV. Choices are: 0, N, No, 1, Y, Yes.
decimal?	CertCapacity	("Certified Capacity", dtDouble)
		The certified capacity for a PSV.
string	CertCapacityUnit	("Certified Capacity Unit", dtString, Maximum Length 15, UoM Factors Unit [Capacity])
		From a lookup list, select the unit of measure for the certified capacity of a safety valve.
decimal?	ColdPressure	("Cold Press", dtDouble, Unit-Based Field)
		The cold differential set pressure for a PSV. (The pressure where a PSV is set while on a test rig using a test fluid at ambient temperature)
bool?	Corrosive	("Corrosive Service", dtInteger)
		Indicate whether the PSV is in corrosive service. Choices are: 0, N, No, 1, Y, Yes.
string	CriticalityServiceCode	("Criticality/Service Code", dtString, Maximum Length 25)
		From a lookup list, select the criticality or service code for a safety valve.
string	CWFilter	("c/w Filter", dtString, Maximum Length 25)
		The type of filter for a PSV if the accessories include a filter.
string	DatasheetNum	("Datasheet No", dtString, Maximum Length 25)
		The datasheet number for a PSV.
string	DesignType	("Design Type", dtString, Maximum Length 25)
		From a lookup list, select the type of design for a safety valve. Examples: Conventional, Pilot-operated.
string	DischargesTo	("Discharges To", dtString, Maximum Length 25)
		From a lookup list, select where the safety valve discharges to.
string	Elastomers	("Elastomers", dtString, Maximum Length 25)
		The type of elastomer for a PSV.
bool?	Erosive	("Erosive Service", dtInteger)
		Indicate whether the PSV is in erosive service. Choices are: 0, N, No, 1, Y, Yes.
string	FABodyMaterial	("FA Body Material", dtString, Maximum Length 15)

		From a lookup list, select the type of material for the body of the flame arrestor for a safety valve.
string	FAOrientation	("FA Orientation", dtString, Maximum Length 15)
		From a lookup list, select the orientation of the flame arrestor for a safety valve.
string	Fire	("Sizing Basis", dtString, Maximum Length 25)
		From a lookup list, select the type of fire protection for a safety valve.
bool?	FlameArrestor	("Flame Arrestor", dtInteger)
		Indicate whether a flame arrestor is present for a PSV. Choices are: 0, N, No, 1, Y, Yes.
decimal?	FlashWt	("Flash Point", dtDouble, Unit-Based Field)
		The flash point temperature for a PSV.
string	FluidPassingStatus	("Fluid Passing StatE", dtString, Maximum Length 20)
		From a lookup list, selec the fluid passing status for a safety valve. Choices: Liquid, Gas.
string	Gag	("Gag", dtString, Maximum Length 25)
		The type of gag for a PSV.
decimal?	GasCompress	("Compressibility (Z)", dtDouble)
		The compressibility factor for a PSV.
decimal?	GasMolecularWt	("Gas Molecular Wt (M)", dtDouble)
		The molecular weight of the gas process for a PSV.
string	GasOther	("Gas Other", dtString, Maximum Length 25)
		Enter any other gas process condition(s) for a PSV.
decimal?	GasSpHeat	("Specific Heat Ratio (k)", dtDouble)
		The specific heat ratio for a PSV.
decimal?	GasTemp	("Gas Temp", dtDouble, Unit-Based Field)
		The temperature of the gas process for a PSV.
string	Guide	("Guide", dtString, Maximum Length 25)
		From a lookup list, select the type of guide for a safety valve.
bool?	InletBlockValve	("Inlet Block Valve", dtInteger)
		Indicate whether an inlet block valve is present for a PSV. Choices are: 0, N, No, 1, Y, Yes.
string	InletConnection	("Inlet Connection End", dtString, Maximum Length 25)
		From a lookup list, select the type of connection on the inlet for a safety valve.
decimal?	InletSize	("Inlet Size", dtDouble, Unit-Based Field)
		The inlet size for a PSV.
string	InOpenMech	("Inlet-Open Mechanism", dtString, Maximum Length 15)
		From a lookup list, select the open mechanism on the inlet for a safety valve.
decimal?	LeakRate	("Leakage Rate Area", dtDouble)
		Enter the leakage rate for a safety valve.

string	LeakRateUnit	("Leakage Rate Unit", dtString, Maximum Length 15, UoM Factors Unit [Leakage Rate])
		From a lookup list, select the unit of measure for the leakage rate for a safety valve.
string	LevelPlain	("Lever Plain", dtString, Maximum Length 25)
		The type of plain lever for a PSV.
string	LeverPacked	("Lever Packed", dtString, Maximum Length 25)
		The type of lever packed for a PSV.
string	Location	("Location Notes", dtString, Maximum Length 4000)
		A descriptive explanation for the physical location of the PSV.
string	LocationID	("Location ID", dtString, Maximum Length 25)
		From a lookup list, select the physical location tag applicable to the safety valve configuration.
string	Material	("Body Material", dtString, Maximum Length 15)
		From a lookup list, select the type of material for a safety valve.
decimal?	MaxBpRating	("Max BP Rating", dtDouble, Unit-Based Field)
		The maximum back pressure rating for a PSV.
decimal?	MolecularWt	("Molecular Weight", dtDouble)
		The molecular weight for a PSV.
string	NozzleAndDisc	("Nozzle and Disc", dtString, Maximum Length 25)
		From a lookup list, select the type of material of the nozzle and disc for a safety valve.
string	OilOther	("Liquid Other", dtString, Maximum Length 25)
		Enter any other oil process condition(s) for a PSV.
decimal?	OricCalcArea	("Orif-Calculated Area", dtDouble)
		Enter the orifice calculated area for a safety valve.
string	OrifCalcAreaUnit	("Orif-Calc Area Unit", dtString, Maximum Length 15, UoM Factors Unit [Area])
		From a lookup list, select the unit of measure for the orifice calculated area for a safety valve.
string	OrificeDesignation	("Orifice Designation", dtString, Maximum Length 25)
		From a lookup list, select the orifice designation for a safety valve.
decimal?	OrifSelArea	("Orif-Selected Area", dtDouble)
		Enter the orifice selected area for a safety valve.
string	OrifSelAreaUnit	("Orif-Sel Area Unit", dtString, Maximum Length 15, UoM Factors Unit [Area])
		From a lookup list, select the unit of measure for the orifice selected area for a safety valve.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter PSV equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all PSV static data forms.

		Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50) Enter PSV equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all PSV static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	OutCarsealNum	("Outlet Carseal No", dtString, Maximum Length 25) The outlet carseal number for a PSV.
bool?	OutCarsealOpen	("Outlet Carseal Locked Open", dtInteger) Indicate whether the outlet carseal is locked open for a PSV. Choices are: 0, N, No, 1, Y, Yes.
bool?	OutletBlockValve	("Outlet Block Valve", dtInteger) Indicate whether an outlet block valve is present for a PSV. Choices are: 0, N, No, 1, Y, Yes.
string	OutletConnection	("Outlet Connection End", dtString, Maximum Length 25) From a lookup list, select the type of connection on the outlet side for a safety valve.
decimal?	OutletSize	("Outlet Size", dtDouble, Unit-Based Field) The outlet size for a PSV.
string	OutlinedDimensions	("Outlined Dimensions", dtString, Maximum Length 25) The outlined dimensions for a PSV.
string	OutOpenMech	("Outlet-Open Mechanism", dtString, Maximum Length 15) From a lookup list, select the open mechanism on the outlet for a safety valve.
decimal?	OverPressure	("Over Press", dtDouble, Unit-Based Field) The overpressure percentage for a PSV. The pressure increase above set pressure when the discharge capacity is attained.
decimal?	PalletAdjPress	("Pallet Adj - Press", dtDouble, Unit-Based Field) The pallet adjustment on the pressure side for a PVRV.
decimal?	PalletAdjVac	("Pallet Adj - Vacuum", dtDouble, Unit-Based Field) The pallet adjustment on the vacuum side for a PVRV.
string	PalletMatPress	("Pallet Material - Press", dtString, Maximum Length 15) From a lookup list, select the type of material of the pallet on the pressure side for a PVRV.
string	PalletMatVac	("Pallet Material - Vacuum", dtString, Maximum Length 15) From a lookup list, select the type of material of the pallet on the vacuum side for a PVRV.
decimal?	PalletWtPress	("Pallet Wt - Press", dtDouble, Unit-Based Field) The pallet weight on the pressure side for a PVRV.
decimal?	PalletWtVac	("Pallet Wt - Vacuum", dtDouble, Unit-Based Field) The pallet weight on the vacuum side for a PVRV.

string	PlainCap	("Plain cap", dtString, Maximum Length 25)
		The type of plain cap for a PSV.
string	PltBodyMat	("Pilot Body Material", dtString, Maximum Length 15)
		From a lookup list, select the type of material for the pilot body of a safety valve.
decimal?	PressureMax	("BP Factor/Max Press", dtDouble, Unit-Based Field)
		The back-pressure factor for a PSV.
decimal?	PressureNormal	("Normal Press", dtDouble, Unit-Based Field)
		The normal pressure for a PSV.
string	PurchaseOrderNum	("Purchase Order", dtString, Maximum Length 25)
		The purchase order number for a PSV.
string	RemotePressurePickup	("Remote Press Pickup", dtString, Maximum Length 25)
		The remote pressure pickup for a PSV.
decimal?	RequiredCapacity	("Required Capacity", dtDouble)
		The required capacity for a PSV.
string	RequiredCapacityUnit	("Required Capacity Unit", dtString, Maximum Length 15, UoM Factors Unit [Capacity])
		From a lookup list, select the unit of measure for the required capacity of a safety valve.
string	RestrictLift	("Restricted Lift", dtString, Maximum Length 25)
		From a lookup list, select the type of restricted lift for a PSV.
string	RevisionNum	("Revision No", dtString, Maximum Length 10)
		The revision number for a PSV.
string	Rings	("Rings", dtString, Maximum Length 25)
		From a lookup list, select the type of rings for a safety valve.
string	SealMatVac	("Seal Material - Vacuum", dtString, Maximum Length 15)
		From a lookup list, select the type of material for the seal on the vacuum side of a PVRV.
string	SealPress	("Seal Material - Press", dtString, Maximum Length 15)
		From a lookup list, select the type of material for the seal on the pressure side of a PVRV.
string	SealType	("Seat Type", dtString, Maximum Length 25)
		From a lookup list, select the type of seat for a PSV.
decimal?	SetPressure	("Set Press", dtDouble, Unit-Based Field)
		The set pressure for a PSV. The pressure at the inlet where the SRV commences to lift while in service.
decimal?	SetPressVac	("Set Press - Vacuum", dtDouble, Unit-Based Field)
		The vacuum set pressure for a PVRV.
bool?	Shellside	("Shellside", dtInteger)
		Indicate whether the PSV is for the shell side. Choices are: 0, N, No, 1, Y, Yes.
string	SizeCalcCode	("Calculation Code", dtString, Maximum Length 20)

		From a lookup list, select the code used to calculate the sizing of the safety valve.
string	SoftSeatMat	("Soft Seat Material", dtString, Maximum Length 25)
		From a lookup list, select the type of material for the soft seat of a safety valve.
decimal?	SpecGravity	("Spec Gravity", dtDouble)
		The specific gravity for a PSV.
string	SpecialClean	("Special Cleaning", dtString, Maximum Length 5)
		Indicate whether a special cleaning is required for a safety valve.
string	SpringAluminized	("Spring Aluminized", dtString, Maximum Length 25)
		The aluminized spring for a PSV.
string	SpringMaterial	("Spring Material", dtString, Maximum Length 15)
		From a lookup list, select the type of material of the spring for a safety valve.
string	SpringNumber	("Spring Number", dtString, Maximum Length 25)
		The spring number for a PSV.
int?	SpringRange	("Spring Range", dtInteger)
		The spring range for a PSV.
decimal?	SpringSetPressure	("Spring Set Press", dtDouble, Unit-Based Field)
		The spring set pressure for a PSV.
decimal?	StmSaturationTemp	("Saturation Temp", dtDouble, Unit-Based Field)
		The saturation temperature of the steam process for a PSV.
decimal?	StmSuperHeat	("Superheat Factor", dtDouble)
		The superheat percentage of the steam process for a PSV.
decimal?	StmTemp	("Stm Temp", dtDouble, Unit-Based Field)
		The temperature of the steam process for a PSV.
decimal?	SuperImposedBp	("Super Imposed BP", dtDouble, Unit-Based Field)
		The superimposed back pressure (the outlet pressure caused by the flow through the valve) for a PSV.
decimal?	TempMax	("Max Temp", dtDouble, Unit-Based Field)
		The maximum temperature for a PSV.
decimal?	TempNormal	("Normal Temp", dtDouble, Unit-Based Field)
		The normal temperature for a PSV.
bool?	Toxic	("Toxic Service", dtInteger)
		Indicate whether the PSV is in toxic service. Choices are: 0, N, No, 1, Y, Yes.
string	TrimOther	("Trim Other", dtString, Maximum Length 25)
		Enter any additional type of trim for a PSV.
bool?	Tubeside	("Tubeside", dtInteger)
		Indicate whether the PSV is for the tube side. Choices are: 0, N, No, 1, Y, Yes.
string	TypeFacingInlet	("Inlet Facing Type", dtString, Maximum Length 25)
		From a lookup list, select the type of facing on the inlet side for a safety valve.

string	TypeFacingOutlet	("Outlet Facing Type", dtString, Maximum Length 25)
		From a lookup list, select the type of facing on the outlet side for a safety valve.
bool?	UD1	("UD-1", dtInteger)
		Indicate whether the manufacturer/assembler certificate of conformance for Rupture Disk Devices will be attached. Choices are: 0, N, No, 1, Y, Yes.
bool?	UV1	("UV-1", dtInteger)
		Indicate whether the manufacturer/assembler certificate of conformance for Pressure Relief Valves will be attached. Choices are: 0, N, No, 1, Y, Yes.
string	Vendor	("Vendor", dtString, Maximum Length 25)
		The name of the vendor for a PSV.
string	VentWithBugScreen	("Vent with Bugscreen", dtString, Maximum Length 25)
		The type of vent for a PSV.
decimal?	Viscosity	("Viscosity", dtDouble, Unit-Based Field)
		The viscosity for a PSV.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment
PSVAccess	lstPSVAccess

RE_GPTurbStat (rfRE_GPTurbStat)

General Purpose Steam Turbine Static: the static information associated with equipment types derived from the GP Stm Turbine supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
decimal?	AllowSpeedMax	("Allow Speed Max", dtDouble, Unit-Based Field)
		The highest speed, in rpm, at which the manufacturer design will permit continuous operation.
decimal?	AllowSpeedMin	("Allow Speed Min", dtDouble, Unit-Based Field)
		The lowest speed, in rpm, at which the manufacturer design will permit continuous operation.
string	EndSeals	("End Seals", dtString, Maximum Length 20)
		From a lookup list, select the type of end seals for a turbine.
decimal?	ExhaustHydrotest	("Exhaust Hydrotest", dtDouble, Unit-Based Field)
		The hydrotest pressure of the casing exhaust for a turbine.
decimal?	ExhaustMaxPress	("Exhaust Max Press", dtDouble, Unit-Based Field)
		The highest exhaust steam pressure that the purchaser requires the casing to contain, with steam supplied at maximum inlet conditions.
decimal?	ExhaustMaxTemp	("Exhaust Max Temp", dtDouble, Unit-Based Field)
		The maximum allowable temperature of the casing exhaust for a turbine.
decimal?	ExhaustPressMin	("Stm Exhaust Press Min", dtDouble, Unit-Based Field)

		The lowest exhaust steam pressure at which the turbine is required to operate continuously.
decimal?	ExhaustPressNorm	("Stm Exhaust Press Norm", dtDouble, Unit-Based Field) The exhaust pressure at which the equipment will usually operate. These conditions are the ones at which the highest efficiency is desired.
decimal?	ExhaustTempNorm	("Exhaust Temp Norm", dtDouble, Unit-Based Field) The exhaust temperature at which the equipment will usually operate. These conditions are the ones at which the highest efficiency is desired.
string	ExtLubeSystem	("Ext Lube System", dtString, Maximum Length 20) From a lookup list, select the type of external lubrication system for a turbine.
decimal?	FirstCritSpeed	("First Critical Speed", dtDouble, Unit-Based Field) The first critical speed, in rpm, at which the system may be in resonance.
decimal?	InletHydrotest	("Inlet Hydrotest", dtDouble, Unit-Based Field) The hydrotest pressure of the casing inlet for a turbine.
decimal?	InletMaxPress	("Inlet Max Press", dtDouble, Unit-Based Field) The maximum allowable pressure of the casing inlet for a turbine.
decimal?	InletMaxTemp	("Inlet Max Temp", dtDouble, Unit-Based Field) The maximum allowable temperature of the casing inlet for a turbine.
decimal?	InletPressMin	("Stm Inlet Press Min", dtDouble, Unit-Based Field) The lowest inlet steam pressure at which the turbine is required to operate continuously.
decimal?	InletPressNorm	("Stm Inlet Press Norm", dtDouble, Unit-Based Field) The inlet pressure at which the equipment will usually operate. These conditions are the ones at which the highest efficiency is desired.
decimal?	InletTempMin	("Stm Inlet Temp Min", dtDouble, Unit-Based Field) The lowest inlet steam temperature at which the turbine is required to operate continuously.
decimal?	InletTempNorm	("Stm Inlet Temp Norm", dtDouble, Unit-Based Field) The inlet temperature at which the equipment will usually operate. These conditions are the ones at which the highest efficiency is desired.
string	InterstageSeals	("Interstage Seals", dtString, Maximum Length 20) From a lookup list, select the type of interstage seals for a turbine.
decimal?	MaxContSpeed	("Max Cont Speed", dtDouble, Unit-Based Field) The speed, in rpm, at least equal to 105 percent of the highest speed required by any of the specified operating conditions.
int?	NumOfStages	("Num of Stages", dtInteger) The number of stages that a pump accommodates.
string	Other5	("Other 5", dtString, Maximum Length 50) Enter GP steam turbine equipment type-specific data. The field can be renamed in the Administration module and will appear on all GP steam turbine static datasheets. Note: The field label may have been renamed in the Visions Administrator > Site Settings.

string	Other6	("Other 6", dtString, Maximum Length 50) Enter GP steam turbine equipment type-specific data. The field can be renamed in the Administration module and will appear on all GP steam turbine static datasheets. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
decimal?	PotMaxPower	("Potential Max Power", dtDouble, Unit-Based Field) The approximate maximum power to which the turbine can be uprated at the specified normal speed and steam conditions when it is furnished with suitable (larger or additional) nozzles and, possibly, with a larger governor-controlled valve or valves.
decimal?	PowerNormal	("Normal Power", dtDouble, Unit-Based Field) The normal power operating condition for a turbine.
decimal?	PowerRated	("Rated Power", dtDouble, Unit-Based Field) The greatest turbine power specified for its corresponding speed. It includes all of the margin required by the driven-equipment specifications.
string	RadialBearings	("Radial Bearings", dtString, Maximum Length 20) From a lookup list, select the type of radial bearings for a turbine.
int?	RelClassID	("Reliability Class", dtInteger) A system lookup list, built from the Reliability Codes, for the available Class of the asset. Only lists Reliability Codes where CodeType = Class.
string	RotatFacGovEnd	("Rotation Facing Gov End", dtString, Maximum Length 20) From a lookup list, select the direction for the governor drive rotation of a turbine.
string	Rotor	("Rotor", dtString, Maximum Length 20) From a lookup list, select the type of rotor in the turbine.
string	Shaft	("Shaft", dtString, Maximum Length 20) From a lookup list, select the type of shaft for a turbine.
string	SiteLocation	("Site Location", dtString, Maximum Length 35) From a lookup list, select the site location condition for the pump, per API 611 section 4.1.6.
string	SpeedChanger	("Speed Changer", dtString, Maximum Length 20) From a lookup list, select the type of speed changer for a turbine.
decimal?	SpeedNormal	("Normal Speed", dtDouble, Unit-Based Field) The normal speed operating condition for a turbine.
decimal?	SpeedRated	("Rated Speed", dtDouble, Unit-Based Field) The greatest turbine speed specified for its corresponding power. It includes all of the margin required by the driven-equipment specifications.
string	StmControlManuf	("Stm Control Manufacturer", dtString, Maximum Length 50) From a lookup list, select the manufacturer for the speed changer of a turbine.
string	StmControlModelNo	("Stm Control Model No", dtString, Maximum Length 35) The model number for the speed changer of a turbine.
decimal?	StmMaxExhaust	("Stm Max Exhaust", dtDouble, Unit-Based Field)

		The maximum exhaust steam rate performance for a turbine.
decimal?	StmMinInlet	("Stm Min Inlet", dtDouble, Unit-Based Field)
		The minimum inlet steam rate performance for a turbine.
decimal?	StmRateNorm	("Stm Rate Norm", dtDouble, Unit-Based Field)
		The normal or certified steam rate performance for a turbine.
decimal?	StmRateRated	("Stm Rate Rated", dtDouble, Unit-Based Field)
		The rated steam rate performance for a turbine.
string	ThrustBearings	("Thrust Bearings", dtString, Maximum Length 20)
		From a lookup list, select the type of thrust bearings for a turbine.
decimal?	TripSpeed	("Trip Speed", dtDouble, Unit-Based Field)
		The speed, in rpm, at which the independent emergency overspeed device operates to shut down the turbine. The trip speed setting will vary with the class of governor.
string	TripValve	("Trip Valve", dtString, Maximum Length 20)
		From a lookup list, select the type of trip valve for a turbine.
string	UnusualCond	("Unusual Conditions", dtString, Maximum Length 20)
		From a lookup list, select the unusual site condition for the turbine, per API 611 section 4.1.6.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

RE_MotorStat (rfRE_MotorStat)

Motor Static: the static information associated with equipment types derived from the RE Motor supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
decimal?	Amps	("Amps", dtDouble, Unit-Based Field)
		The rated full-load amperage, in amps, for the motor.
decimal?	Efficiency	("Efficiency", dtDouble, Unit-Based Field)
		The full-load efficiency percentage for the motor.
string	Enclosure	("Enclosure", dtString, Maximum Length 10)
		From a lookup list, select the type of enclosure for the motor. Examples: ODP, TENV, EXP.
string	FrameSize	("Frame Size", dtString, Maximum Length 20)
		From a lookup list, select the NEMA frame size for the motor.
decimal?	Hertz	("Hertz", dtDouble, Unit-Based Field)
		The frequency, in Hz, at which the motor cycles.
decimal?	HP	("HP", dtDouble, Unit-Based Field)
		The rated horsepower for the motor.
string	InsulationClass	("Insulation Class", dtString, Maximum Length 10)

		From a lookup list, select the insulation class for the motor.
string	LockedRotorCode	("Locked-Rotor Code", dtString, Maximum Length 10)
		From a lookup list, select the locked-rotor or design code for the motor.
string	MotorType	("Motor Type", dtString, Maximum Length 20)
		From a lookup list, select the type of motor. Examples: Synchronous, Squirrel Cage, Direct Current, Shunt-Wound.
string	MountType	("Mount Type", dtString, Maximum Length 20)
		From a lookup list, select the type of mount for the motor. Examples: Rigid base, Resilient base.
string	NEMADesign	("NEMA Design", dtString, Maximum Length 10)
		From a lookup list, select the NEMA design code for the motor.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter GP steam turbine equipment type-specific data. The field can be renamed in the Administration module and will appear on all GP steam turbine static datasheets. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter GP steam turbine equipment type-specific data. The field can be renamed in the Administration module and will appear on all GP steam turbine static datasheets. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Phase	("Phase", dtString, Maximum Length 20)
		From a lookkup list, select the phase type for the motor. Examples: Single, 3-phase
int?	RelClassID	("Reliability Class", dtInteger)
		A system lookup list, built from the Reliability Codes, for the available Class of the asset. Only lists Reliability Codes where CodeType = Class.
string	Rotation	("Rotation", dtString, Maximum Length 20)
		From a lookup list, select the direction of the rotation of the motor; typically clockwise or counter-clockwise..
decimal?	RPM	("RPM", dtDouble, Unit-Based Field)
		The rated full-load speed, in rpm, for the motor.
decimal?	ServiceFactor	("Service Factor", dtDouble)
		The service factor for the motor.
string	SiteLocation	("Site Location", dtString, Maximum Length 35)
		From a lookup list, select the site location condition for the pump, per API 611 section 4.1.6.
string	TimeRating	("Time Rating", dtString, Maximum Length 20)
		From a lookup list, select the time rating for the motor.
decimal?	Torque	("Torque", dtDouble, Unit-Based Field)
		The full-load torque for the motor.
string	UnusualCond	("Unusual Conditions", dtString, Maximum Length 20)

		From a lookup list, select the unusual site condition for the turbine, per API 611 section 4.1.6.
decimal?	Voltage	("Voltage", dtDouble, Unit-Based Field) The rated voltage for the motor.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

RE_PumpStat (rfPumpStat)

Centrifugal Pump Static: the static information associated with equipment types derived from the Centrifugal Pump supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
decimal?	AllowRegionFrom	("Allow Oper Region/From", dtDouble, Unit-Based Field) The starting range of the allowable operating region (the hydraulic coverage over which the pump is allowed to operate) for the pump.
decimal?	AllowRegionTo	("Allow Oper Region/To", dtDouble, Unit-Based Field) The starting range of the allowable operating region (the hydraulic coverage over which the pump is allowed to operate) for the pump.
decimal?	BaseplateWt	("Baseplate Wt", dtDouble, Unit-Based Field) The weight or mass of the baseplate of the pump.
string	CaseMaterial	("Case Material", dtString, Maximum Length 15) From a lookup list, select the type of material for the case of the pump.
string	CasingMount	("Casing Mounting", dtString, Maximum Length 15) From a lookup list, select the type of casing mounting for the pump.
string	CasingSplit	("Casing Split", dtString, Maximum Length 15) From a lookup list, select the type of split in the casing of the pump.
string	CasingType	("Casing Type", dtString, Maximum Length 15) From a lookup list, select the type of casing for the pump.
decimal?	ChlorConc	("Chloride Concentration", dtDouble, Unit-Based Field) The concentration of chlorides present in the pumped fluid of the pump.
decimal?	DiffHead	("Differential Head", dtDouble, Unit-Based Field) The operating differential head of the pump, per API section 5.1.3.
decimal?	DiffPress	("Differential Press", dtDouble, Unit-Based Field) The operating differential pressure of the pump, per API section 5.1.3.
string	Diffusers	("Diffusers", dtString, Maximum Length 15) From a lookup list, select the type of material for the diffusers of the pump.
decimal?	DischPress	("Discharge Press", dtDouble, Unit-Based Field) The operating discharge pressure of the pump, per API section 5.1.3.
string	DriverType	("Driver Type", dtString, Maximum Length 20) From a lookup list, select the type of driver for the pump.

decimal?	DriverWt	("Driver Wt", dtDouble, Unit-Based Field)
		The weight or mass of the driver.
decimal?	Efficiency	("Efficiency (%)", dtDouble)
		The performance efficiency percentage for the pump.
bool?	Flammable	("Flammable", dtInteger)
		Indicate whether the liquid in the pump is flammable. Choices are: 0, N, No, 1, Y, Yes.
decimal?	GearWt	("Gear Wt", dtDouble, Unit-Based Field)
		The weight or mass of the gear.
decimal?	H2SConc	("H2S Concentration", dtDouble, Unit-Based Field)
		The amount of wet H2S that may be present in the pumped fluid of the pump.
bool?	Hazardous	("Hazardous", dtInteger)
		Indicate whether the liquid in the pump is hazardous. Choices are: 0, N, No, 1, Y, Yes.
decimal?	ImpDiaMax	("Impeller Dia Max", dtDouble, Unit-Based Field)
		The maximum impeller diameter for the pump.
decimal?	ImpDiaMin	("Impeller Dia Min", dtDouble, Unit-Based Field)
		The minimum impeller diameter for the pump.
decimal?	ImpDiaRated	("Impeller Dia Rated", dtDouble, Unit-Based Field)
		The rated impeller diameter for the pump.
string	ImpellerType	("Impeller Type", dtString, Maximum Length 15)
		From a lookup list, select the type of impeller for the pump.
string	ImpMaterial	("Impeller Material", dtString, Maximum Length 15)
		From a lookup list, select the type of material for the impeller of the pump.
string	LiquidType	("Liquid Type", dtString, Maximum Length 20)
		From a lookup list, select the type of liquid for the pump, per API 610 section 5.1.3.
string	MaterialClass	("Material Class", dtString, Maximum Length 15)
		From a lookup list, select the material class for the pump parts. Per API 610, Annex H.
decimal?	MaxHeadImp	("Max Head - Rated Imp", dtDouble, Unit-Based Field)
		The maximum head at the rated impeller for the pump.
decimal?	MaxPowerImp	("Max Power - Rated Imp", dtDouble, Unit-Based Field)
		The maximum power at the rated impeller for the pump.
decimal?	MDMT	("MDMT", dtDouble, Unit-Based Field)
		The lowest mean metal temperature (through the thickness) expected in service of the surrounding environment.
decimal?	MinContFlowStb	("Min Cont Flow Stable", dtDouble, Unit-Based Field)
		The lowest flow at which the pump can operate without exceeding the vibration limits.
decimal?	MinContFlowThr	("Min Cont Flow Thermal", dtDouble, Unit-Based Field)
		The lowest flow at which the pump can operate without its operation being impaired by the temperature rise of the pumped liquid.

bool?	NACECompliant	("NACE Compliant", dtInteger) Indicate whether reduced-hardness materials are required, in accordance with NACE MR0175. Choices are: 0, N, No, 1, Y, Yes.
decimal?	NormalFlow	("Normal Flow", dtDouble, Unit-Based Field) The normal operating flow of the pump, per API section 5.1.3
decimal?	NPSHA	("NPSHA", dtDouble, Unit-Based Field) The net positive suction head available for the pump, per API section 5.1.3.
decimal?	NPSHR_Max	("NPSHR - Max", dtDouble, Unit-Based Field) The net positive suction head required at maximum flow for the pump.
decimal?	NPSHR_Rated	("NPSHR - Rated", dtDouble, Unit-Based Field) The net positive suction head required at rated flow for the pump.
int?	NumOfStages	("Num of Stages", dtInteger) The number of stages that a pump accommodates.
string	Other5	("Other 5", dtString, Maximum Length 50) Enter centrifugal pump equipment type-specific data. The field can be renamed in the Administration module and will appear on all centrifugal pump static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50) Enter centrifugal pump equipment type-specific data. The field can be renamed in the Administration module and will appear on all centrifugal pump static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
decimal?	PowerRated	("Rated Power", dtDouble, Unit-Based Field) The rated power for the pump.
decimal?	PrefRegionFrom	("Pref Oper Region/From", dtDouble, Unit-Based Field) The starting range of the preferred operating region (the hydraulic coverage over which the vibration is within the base limit) for the pump.
decimal?	PrefRegionTo	("Pref Oper Region/To", dtDouble, Unit-Based Field) The ending range of the preferred operating region (the hydraulic coverage over which the vibration is within the base limit) for the pump.
string	PropCurveNo	("Proposal Curve No", dtString, Maximum Length 25) The performance or characteristic curve number for the pump.
decimal?	PumpTempMax	("Pumping Temp Max", dtDouble, Unit-Based Field) The maximum pumping temperature of the liquid for the pump.
decimal?	PumpTempMin	("Pumping Temp Min", dtDouble, Unit-Based Field) The minimum pumping temperature of the liquid for the pump.
decimal?	PumpTempNorm	("Pumping Temp Norm", dtDouble, Unit-Based Field) The normal pumping temperature of the liquid for the pump.
decimal?	PumpWt	("Pump Wt", dtDouble, Unit-Based Field) The weight or mass of the pump.

decimal?	RatedFlow	("Rated Flow", dtDouble, Unit-Based Field) The rated operating flow of the pump, per API section 5.1.3.
int?	RelClassID	("REL CLASS ID", dtInteger) A system generated counter identifying the reliability code class for the pump.
decimal?	RelDensMax	("Rel Density Max", dtDouble, Unit-Based Field) The maximum relative density of the liquid for the pump.
decimal?	RelDensMin	("Rel Density Min", dtDouble, Unit-Based Field) The minimum relative density of the liquid for the pump.
decimal?	RelDensNorm	("Rel Density Norm", dtDouble, Unit-Based Field) The normal relative density of the liquid for the pump.
string	Rotation	("Rotation", dtString, Maximum Length 20) From a lookup list, select the direction of the rotation of the pump; typically clockwise or counter-clockwise.
string	SealMaterial	("Seal Material", dtString, Maximum Length 15) From a lookup list, select the type of material for the seal of the pump.
decimal?	SealSize	("Seal Size", dtDouble, Unit-Based Field) The size of the seal for the pump.
string	SealType	("Seal Type", dtString, Maximum Length 15) From a lookup list, select the type of seal for the pump.
string	ShaftMaterial	("Shaft Material", dtString, Maximum Length 15) From a lookup list, select the type of material for the shaft of the pump.
string	SiteLocation	("Site Location", dtString, Maximum Length 35) From a lookup list, select the site location condition for the pump, per API section 5.1.30
decimal?	SpecHeat	("Specific Heat", dtDouble, Unit-Based Field) The specific heat of the liquid for the pump.
decimal?	Speed	("Speed", dtDouble, Unit-Based Field) The speed of the pump.
decimal?	SucPressMax	("Suction Press Max", dtDouble, Unit-Based Field) The maximum operating suction pressure of the pump, per API section 5.1.3.
decimal?	SucPressRated	("Suction Press Rated", dtDouble, Unit-Based Field) The rated operating suction pressure of the pump, per API section 5.1.3.
decimal?	TotalWt	("Total Wt", dtDouble, Unit-Based Field) The total weight or mass of each item of the equipment.
string	UnusualCond	("Unusual Conditions", dtString, Maximum Length 20) From a lookup list, select the unusual site condition for the pump, per API 610 section 5.1.30.
decimal?	VapPressMax	("Vapour Press Max", dtDouble, Unit-Based Field) The maximum vapour pressure of the liquid for the pump.
decimal?	VapPressMin	("Vapour Press Min", dtDouble, Unit-Based Field) The minimum vapour pressure of the liquid for the pump.

decimal?	VapPressNorm	("Vapour Press Norm", dtDouble, Unit-Based Field) The normal vapour pressure of the liquid for the pump.
decimal?	ViscMax	("Viscosity Max", dtDouble, Unit-Based Field) The maximum viscosity of the liquid for the pump.
decimal?	ViscMin	("Viscosity Min", dtDouble, Unit-Based Field) The minimum viscosity of the liquid for the pump.
decimal?	ViscNorm	("Viscosity Norm", dtDouble, Unit-Based Field) The normal viscosity of the liquid for the pump.
string	WearRings	("Wear Rings", dtString, Maximum Length 15) From a lookup list, select the type of material for the wear rings of the pump.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

StackStat (rfStackStat)

Stack Static: the static information associated with equipment types derived from the Stack supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
decimal?	GuyWireDiameter	("Guy Wire Diameter", dtDouble, Unit-Based Field) The diameter of a guy wire for a stack.
string	GuyWireMaterial	("Guy Wire Material", dtString, Maximum Length 15) From a lookup list, select the type of material that the guy wire is constructed from.
int?	GuyWireNumber	("Guy Wire Number", dtInteger) The number of similar guy wires for a stack.
decimal?	MaxFlowRate	("Max Flow Rate", dtDouble, Unit-Based Field) The maximum flow rate for a stack.
string	Other5	("Other 5", dtString, Maximum Length 50) Enter stack equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all stack static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50) Enter stack equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all stack static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

ST_CraneStat (rfST_CraneStatVessel)

Structures Crane Static: the static information associated with equipment types derived from the ST Crane supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	BoomType	("Boom Type", dtString, Maximum Length 35)
		From a lookup list, select the type of boom for the crane.
string	CraneBaseType	("Crane Base Type", dtString, Maximum Length 35)
		From a lookup list, select the type of base for the crane.
decimal?	CraneMaxLoad	("Crane Max Load", dtDouble, Unit-Based Field)
		The maximum load for the crane.
decimal?	MaxOperXAxis	("Max Oper Movement X Axis ", dtDouble, Unit-Based Field)
		The X axis movement allowed by the crane.
decimal?	MaxOperYAxis	("Max Oper Movement Y Axis ", dtDouble, Unit-Based Field)
		The Y axis movement allowed by the crane.
decimal?	MaxRadius	("Crane Max Radius ", dtDouble, Unit-Based Field)
		The maximum distance radius of the crane.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter ST crane equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all ST Crane static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter ST crane equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all ST Crane static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

ST_FixedUWStat (rfSTFixedUWStat)

Structures Fixed Underwater Static: the static information associated with equipment types derived from the ST Fixed Underwater supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
bool?	CorrProtection	(“Corrosion Protection”, dtInteger)
		Indicate whether the structure has corrosion protection. Choices are: 0, N, No, 1, Y, Yes.

bool?	GroutedPIles	(“Grouted PIles”, dtInteger) Indicate whether the piles are grouted on the structure. Choices are: 0, N, No, 1, Y, Yes.
decimal?	Height	("Fixed U/W Height", dtDouble, Unit-Based Field) The height of the fixed underwater structure. This can be obtained from the ASME Data Report, the manufacturer drawings or engineering specifications.
decimal?	Length	("Fixed U/W Length", dtDouble, Unit-Based Field) The length of the fixed underwater structure. This can be obtained from the ASME Data Report, the manufacturer drawings or engineering specifications.
string	LongFraming	("Long Framing", dtString, Maximum Length 35) From a lookup list, select the style of long framing of the fixed underwater structure.
bool?	MudMat	("Mud Mat", dtInteger) Indicate whether the structure has mud mats. Choices are: 0, N, No, 1, Y, Yes.
int?	NumBays	("Num of Bays", dtInteger) The number of bays on the structure.
int?	NumLegPiles	("Num of Leg Piles", dtInteger) The number of leg piles on the structure.
int?	NumLevels	("Num of Levels", dtInteger) The number of levels on the structure.
int?	NumSkirtPiles	("Num of Skirt Piles", dtInteger) The number of skirt piles on the structure.
string	Other5	("Other 5", dtString, Maximum Length 50) Enter ST fixed underwater equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all ST fixed underwater static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50) Enter ST fixed underwater equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all ST fixed underwater static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	SoilType	("Soil Type", dtString, Maximum Length 35) From a lookup list, select the soil type that the fixed underwater structure rests on.
decimal?	StructJktWeight	("Structure/Jacket Weight", dtDouble, Unit-Based Field) The length of the fixed underwater structure. This can be obtained from the ASME Data Report, the manufacturer drawings or engineering specifications.
bool?	SZProtect	("Splash Zone Protection", dtInteger)

		Indicate whether the structure has Splash Zone Protection. Choices are: 0, N, No, 1, Y, Yes.
string	TransverseFraming	("Transverse Framing", dtString, Maximum Length 35) From a lookup list, select the style of transverse framing of the fixed underwater structure.
decimal?	WaterDepth	("Water Depth", dtDouble, Unit-Based Field) The water depth of the fixed underwater structure.
decimal?	Width	("Fixed U/W Width", dtDouble, Unit-Based Field) The width of the fixed underwater structure. This can be obtained from the ASME Data Report, the manufacturer drawings or engineering specifications.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

ST_HelipadStat (rfST_HelipadStat)

Structures Helipad Static: the static information associated with equipment types derived from the ST Helipad supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
decimal?	DeckLength	("Deck Length", dtDouble, Unit-Based Field) The length of the deck for the helipad.
decimal?	DeckWeight	("Deck Weight", dtDouble, Unit-Based Field) The weight of the deck for the helipad.
decimal?	HeliMaxGW	("Heli Max Gross Weight", dtDouble, Unit-Based Field) The maximum gross weight of a helicopter permitted to land on the helipad.
decimal?	HelipadCategory	("Helipad Category", dtString, Maximum Length 35) From a lookup list, select the category of helicopter that is permitted to land on the helipad.
string	HelipadType	("Helipad Type", dtString, Maximum Length 20) From a lookup list, select the type of helipad.
string	Other5	("Other 5", dtString, Maximum Length 50) Enter ST helipad equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all ST helipad static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50) Enter ST helipad equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all ST helipad static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

ST_TopsideStat (rfST_TopsideStat)

Structures Topside Static: the static information associated with equipment types derived from the ST Topside supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
decimal?	Height	("Height", dtDouble, Unit-Based Field)
		The height of the topside structure. This can be obtained from the ASME Data Report, the manufacturer drawings or engineering specifications.
decimal?	Length	("Length", dtDouble, Unit-Based Field)
		The length of the topside structure. This can be obtained from the ASME Data Report, the manufacturer drawings or engineering specifications.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter ST topside equipment type-specific data. Since these fields can be renamed in the Administration module, the ST topside will appear on all ST topside static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter ST topside equipment type-specific data. Since these fields can be renamed in the Administration module, the name will appear on all ST topside static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
decimal?	Width	("Width", dtDouble, Unit-Based Field)
		The width of the topside structure. This can be obtained from the ASME Data Report, the manufacturer drawings or engineering specifications.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

TankStatic (rfTankStatic)

Tank Static: the static information associated with equipment types derived from the Tank supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	Agitator	("Agitator", dtString, Maximum Length 20)
		From a lookup list, select the type agitator present in a tank.
string	CircumferentialJoints	("Circumferential Joints", dtString, Maximum Length 20)

		From a lookup list, select the types of circumferential (horizontal) tank plate joints. Example: Butt Weld, Lap Weld, Riveted, etc. This can be obtained from the API Data Report or the construction drawings.
string	ContainType	("Contain Type", dtString, Maximum Length 15)
		From a lookup list, select the type of secondary containment for a tank. Example: Dikes, Double-walled, etc. This can be obtained from the API Data Report or the construction drawings.
decimal?	CoverDepth	("Cover Depth", dtDouble, Unit-Based Field)
		The depth of the soil cover over the ground water monitors for a tank. This can be obtained from the Data Report or design specifications.
DateTime?	DateCompletedReconst	("Date Completed Reconstruction", dtDate)
		Select the date the reconstruction of a tank was completed. This can be obtained from the API Data Report or the nameplate stamping.
decimal?	DesignLiquidLevel	("Design Liquid Level", dtDouble, Unit-Based Field)
		The designed liquid level for a tank. This can be obtained from the API Data Report, the construction drawings or the nameplate stamping.
decimal?	Diameter	("Diameter", dtDouble, Unit-Based Field)
		The diameter for a tank.
string	DikeMaterial	("Dike Material", dtString, Maximum Length 15)
		From a lookup list, select the type of material used in the construction of the dike for a tank. This can be obtained from the construction drawings or an external visual inspection.
string	ErectedBy	("Erected by", dtString, Maximum Length 35)
		The name of the company who completed the field erection of a tank. This can be obtained from the API Data Report or the nameplate stamping.
string	FabricatedBy	("Fabricated by", dtString, Maximum Length 35)
		The name of the company who fabricated the tank. This can be obtained from the API Data Report or the nameplate stamping.
string	FloatingRoofMfg	("Floating Roof Mfg", dtString, Maximum Length 25)
		From a lookup list, select the name of the manufacturer for the floating roof of a tank.
string	FoundationType	("Foundation Type", dtString, Maximum Length 30)
		From a lookup list, select the type of foundation for a tank. Example: Concrete Ring, Gravel, Concrete Pad, etc.
bool?	GrndWtrMonitor	("Ground Water Monitor", dtInteger)
		Identify whether a ground water monitor is present for a tank. This can be obtained from the engineering specifications. Choices are: 0, N, No, 1, Y, Yes.
bool?	InterstitialSpace	("Interstitial Space", dtInteger)
		Identify whether a tank is double-walled with an interstitial space. This can be obtained from the API Data Report or the construction drawings. Choices are: 0, N, No, 1, Y, Yes.
bool?	Ladder	("Ladder", dtInteger)

		Identify whether an external ladder is present for a tank. This can be obtained from the API Data Report or the construction drawings. Choices are: 0, N, No, 1, Y, Yes.
bool?	LeakDetect	("Leak Detection", dtInteger)
		Identify whether leak detection is present for a tank. This can be obtained from the engineering specifications. Choices are: 0, N, No, 1, Y, Yes.
string	LiningType	("Lining Type", dtString, Maximum Length 20)
		From a lookup list, select the type of lining for a tank. This can be obtained from the API Data report, construction drawings or nameplate stamping.
int?	NoOfColumns	("Num of Columns", dtInteger)
		The number of internal roof support columns in a tank. This can be obtained from the API Data Report or the construction drawings.
int?	NoOfCourses	("Num of Courses", dtInteger)
		The total number of shell courses for a tank. This can be obtained from the API Data Report, the construction drawings or the nameplate stamping.
int?	NoOfLegs	("Num of Legs", dtInteger)
		The number of external support legs for a tank. This can be obtained from the API Data Report or the construction drawings.
int?	NumOfManways	("Num of Manways", dtInteger)
		The total number of manways for a tank. This can be obtained from the API Data Report or the construction drawings.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter tank equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all tank static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter tank equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all tank static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
bool?	OverfillProtect	("Ground Water Monitor", dtInteger)
		Identify whether overfill protection is present for a tank. This can be obtained from the engineering specifications. Choices are: 0, N, No, 1, Y, Yes.
string	Product	("Product", dtString, Maximum Length 30)
		From a lookup list, select the product normally contained in the tank. This can be obtained from the API Data Report, the construction drawings or the tank nameplate.
string	ReconstCode	("Reconstruction Code", dtString, Maximum Length 10)
		From a lookup list, select the applicable Code of reconstruction. This may be obtained from the equipment Data Report, engineering specifications or drawings.
string	ReconstructedBy	("Reconstructed By", dtString, Maximum Length 35)

		The name of the company who reconstructed the tank. This can be obtained from the API Data Report, Repair Report or tank nameplate.
string	ReferenceManway	("Reference Manway", dtString, Maximum Length 1)
		The reference manway number of a tank.
string	Region	("Region", dtString, Maximum Length 25)
		From a lookup list, select the region of a tank.
string	Revision	("Revision", dtString, Maximum Length 10)
		The revision of the Code of construction. This can be obtained from the API Data Report, construction drawings or the tank nameplate stamping.
string	RoofSupports	("Roof Supports", dtString, Maximum Length 20)
		From a lookup list, select the type of roof supports of a tank. This can be obtained from the tank data report or design specifications.
bool?	SecContainment	("Sec Containment", dtInteger)
		Identify whether there is secondary leak containment for a tank. This can be obtained from the API Data Report or the construction drawings. Choices are: 0, N, No, 1, Y, Yes.
decimal?	SeismicLoad	("Seismic Load", dtDouble, Unit-Based Field)
		The seismic load capability of a tank.
string	ShellType	("Tank Type", dtString, Maximum Length 30)
		Select the type of tank. Choices are: AST (above ground storage), UST (underground storage). This may be obtained from the API Data Report form, construction drawings or engineering specifications.
decimal?	SizeNominal	("Nominal Size", dtDouble, Unit-Based Field)
		Enter teh nominal size
string	SoilDescription	("Soil description", dtString, Maximum Length 25)
		The type of soil around a tank. Example: Clay, Dirt, Asphalt, etc.). This can be obtained from the engineering specifications.
decimal?	SpecificGravity	("Spec Gravity", dtDouble)
		The specific gravity of the process fluid normally in the tank. This can be obtained from the API Data Report, the construction drawings or the tank nameplate.
bool?	Stairway	("Stairway", dtInteger)
		Identify whether there is an external stairway on the side of the tank. This can be obtained from the API Data Report or the construction drawings. Choices are: 0, N, No, 1, Y, Yes.
decimal?	TankHeight	("Height", dtDouble, Unit-Based Field)
		The overall height of a tank, in appropriate units. This can be be obtained from the ASME Data Report.
string	VerticalJoints	("Vertical Joints", dtString, Maximum Length 20)
		From a lookup list, select the types of vertical tank plate joints of a tank. Example: Butt Weld, Lap Welded, Riveted, etc. This can be obtained from the API Data Report or the construction drawings.
decimal?	WindLoad	("Wind Load", dtDouble, Unit-Based Field)
		The wind load of a tank.

int?	YearReconstructed	("Year Reconstructed", dtInteger) The year of the tank reconstruction. This can be obtained from the API Data Report or the nameplate stamping.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment
TankVolumes	lstTankVolumes

ValveStat(rfValveStat)

Valve Static: the static information associated with equipment types derived from the Valve supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	ActManuf	("Manufacturer", dtString, Maximum Length 50) The manufacturer for the actuator of a valve.
string	ActModelNo	("Model Number", dtString, Maximum Length 50) The model number for the actuator of a valve.
string	ActOrient	("Orientation", dtString, Maximum Length 15) From a lookup list, select the orientation for the actuator of a valve.
decimal?	ActPGMaxPress	("Power Gas Maximum Press", dtDouble, Unit-Based Field) The maximum pressure of the power gas for the actuator of a valve.
decimal?	ActPGNormPress	("Power Gas Normal Press", dtDouble, Unit-Based Field) The normal pressure of the power gas for the actuator of a valve.
decimal?	ActPGSizPress	("Power Gas Sizing Press", dtDouble, Unit-Based Field) The sizing pressure of the power gas for the actuator of a valve.
string	ActPGSrc	("Power Gas Source", dtString, Maximum Length 15) From a lookup list, select the source of the power gas for the actuator of a valve.
string	ActuatorType	("Actuator Type", dtString, Maximum Length 15) From a lookup list, select the type of actuator for a valve.
string	BoltMaterial	("Bolting Material", dtString, Maximum Length 15) From a lookup list, select the type of material for the bolts of a valve.
decimal?	BoltSize	("Flange Bolt Size", dtDouble, Unit-Based Field) The size of the flange bolts for a valve.
string	BonnetMaterial	("Bonnet Material", dtString, Maximum Length 15) From a lookup list, select the type of material for the bonnet of a valve.
string	BonnetType	("Bonnet Type", dtString, Maximum Length 25) From a lookup list, select the type of bonnet for a valve.
string	FailPosition	("Fail Position", dtString, Maximum Length 15) From a lookup list, select the fail position for the actuator of a valve.
bool?	FireSafe	("Fire Safe", dtInteger)

		Identify whether the valve is fire safe. Choices are: 0, N, No, 1, Y, Yes.
string	FlangeBoltMaterial	("Flange Bolt Material", dtString, Maximum Length 15)
		From a lookup list, select the type of material for the flange bolts of a valve.
string	FlangeMaterial	("Flange Material", dtString, Maximum Length 15)
		From a lookup list, select the type of material for the flange of a valve.
string	FlangeType	("Flange Type", dtString, Maximum Length 15)
		From a lookup list, select the type of flange for a valve.
string	FlapPistMaterial	("Flapper/Piston Material", dtString, Maximum Length 15)
		From a lookup list, select the type of material for the flapper or piston of a valve.
decimal?	FlowMax	("Flow Maximum", dtDouble, Unit-Based Field)
		The maximum flow for a valve.
decimal?	FlowNorm	("Flow Normal", dtDouble, Unit-Based Field)
		The normal flow for a valve.
string	FluidPhase	("Fluid Phase", dtString, Maximum Length 25)
		From a lookup list, select the type of fluid phase for a valve. Typically this is gas or vapour.
string	GuideMaterial	("Guide Material", dtString, Maximum Length 15)
		From a lookup list, select the type of material for the guide of a valve.
string	HydFluidType	("Hydraulic Fluid Type", dtString, Maximum Length 15)
		From a lookup list, select the type of hydraulic fluid for the actuator of a valve.
bool?	ManualPump	("Manual Pump", dtInteger)
		Identify whether the actuator of a valve is manual or automatic pump. Choices are: 0, N, No, 1, Y, Yes.
decimal?	MaxStrokeTime	("Maximum Stroke Time", dtDouble, Unit-Based Field)
		The maximum stroke time for the actuator of a valve.
string	NACECompl	("NACE Compliance", dtString, Maximum Length 15)
		From a lookup list, select the NACE paragraph or section applicable to the valve.
int?	NoOfStrokes	("Num of Strokes", dtInteger)
		The number of strokes for the actuator of a valve.
int?	NumOfBolts	("Bolt Count", dtInteger)
		The number of bolts required to mount a valve.
decimal?	OperPressDrop	("Operating Press Drop", dtDouble, Unit-Based Field)
		The operating pressure drop for a valve.
decimal?	OperTempMin	("Operating Temp Minimum", dtDouble, Unit-Based Field)
		The minimum operating temperature for a valve.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter valve equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all valve static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)

		Enter valve equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all valve static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	PackMaterial	("Packing Material", dtString, Maximum Length 15)
		From a lookup list, select the valve packing material.
string	PltAction	("Action", dtString, Maximum Length 15)
		From a lookup list, select the pilot action for a valve.
string	PltDband	("Deadband", dtString, Maximum Length 15)
		From a lookup list, select the pilot deadband for a valve.
decimal?	PltDbandHi	("High Deadband", dtDouble, Unit-Based Field)
		The high setting for the deadband range for the pilot of a valve.
decimal?	PltDbandLo	("Low Deadband", dtDouble, Unit-Based Field)
		The low setting for the deadband range for the pilot of a valve.
decimal?	PltHighSet	("High Setting", dtDouble, Unit-Based Field)
		The high set point for the pilot of a valve.
decimal?	PltLowSet	("Low Setting", dtDouble, Unit-Based Field)
		The low set point for the pilot of a valve.
string	PltManuf	("Manufacturer", dtString, Maximum Length 50)
		The name of the manufacturer for the pilot of a valve
string	PltModelNo	("Model Number", dtString, Maximum Length 50)
		The model number for the pilot of a valve.
bool?	PltPressReqd	("Press Pilot Required", dtInteger)
		Identify whether a pressure pilot is required for a valve. Choices are: 0, N, No, 1, Y, Yes.
decimal?	PltProcConn	("Process Connection Size", dtDouble, Unit-Based Field)
		The size of the process connection for the pilot of a valve.
string	PltProcConnSched	("Process Connection Schedule", dtString, Maximum Length 15)
		From a lookup list, select the schedule of the process connection for the pilot of a valve.
decimal?	PltProcConnThick	("Process Connection Thick", dtDouble, Unit-Based Field)
		The thickness of the process connection for the pilot of a valve.
string	PltReset	("Reset Type", dtString, Maximum Length 15)
		From a lookup list, select reset type for the pilot of a valve. Typically this is manual or automatic.
string	PlugMaterial	("Plug/Ball/Gate Material", dtString, Maximum Length 15)
		From a lookup list, select the type of material for the plug of a valve.
string	SeatMaterial	("Seat Material", dtString, Maximum Length 15)
		From a lookup list, select the type of material for the seat of a valve.
decimal?	ShutoffPress	("Shutoff Press", dtDouble, Unit-Based Field)
		The shut-off pressure for a valve.
string	SlndCloseAction	("Close Valve Action", dtString, Maximum Length 25)

		From a lookup list, select close valve action for the solenoid(s) of a valve.
string	SlndCloseValveTag	("Close Valve Tag Number", dtString, Maximum Length 25)
		The close valve tag number for the solenoid(s) of a valve.
decimal?	SlndCoilVolt	("Coil Voltage", dtDouble, Unit-Based Field)
		The coil voltage for the solenoid(s) of a valve.
string	SlndManuf	("Manufacturer", dtString, Maximum Length 50)
		The manufacturer for the solenoid(s) of a valve.
string	SlndModelNo	("Model Number", dtString, Maximum Length 50)
		The model number for the solenoid(s) of a valve.
string	SlndOpenAction	("Open Valve Action", dtString, Maximum Length 25)
		From a lookup list, select the open valve action for the solenoid(s) of a valve.
string	SlndOpenValveTag	("Open Valve Tag Number", dtString, Maximum Length 25)
		The open valve tag number for the solenoid(s) of a valve.
int?	SlndQuantity	("Quantity Required", dtInteger)
		The quantity of solenoids required for a valve.
string	SlndType	("Type", dtString, Maximum Length 15)
		From a lookup list, select the type of solenoid(s) on a valve.
string	SourSwtPwrGas	("Power Gas Built For", dtString, Maximum Length 15)
		From a lookup list, select whether the actuator for a valve is built for sweet or sour gas.
bool?	SpeedControls	("Speed Controls", dtInteger)
		Identify whether speed controls are on the actuator for a valve. Choices are: 0, N, No, 1, Y, Yes.
string	StemMaterial	("Stem Material", dtString, Maximum Length 15)
		From a lookup list, select the stem material for a valve.
decimal?	TightReqd	("Tightness Required", dtDouble, Unit-Based Field)
		The tightness pressure to be set for a valve.
decimal?	TrimSize	("Trim Size", dtDouble, Unit-Based Field)
		The size of the trim for a valve.
string	TrimType	("Trim Type", dtString, Maximum Length 25)
		From a lookup list, select the type of trim for a valve.
decimal?	ValveStemExt	("Buried Valve Stem Extension", dtDouble, Unit-Based Field)
		The length of the valve stem if it is buried.
string	ValveStemOrient	("Valve Stem Orientation", dtString, Maximum Length 15)
		From a lookup list, select the orientation of the valve stem. Examples:

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment
ValveLimitSwtch	lstValveLimitSwtch

ValveAccess	lstValveAccess

Vessel (rfVessel)

Vessel Static: the static information associated with equipment types derived from the Vessel supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
int?	NumOfCourses	("Num of Courses", dtInteger)
		The number of shell courses for a vessel.
string	Other5	("Other 5", dtString, Maximum Length 50)
		Enter vessel equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all vessel static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	Other6	("Other 6", dtString, Maximum Length 50)
		Enter vessel equipment type-specific data. Since these fields can be re-named in the Administration module, the name will appear on all vessel static data forms. Note: The field label may have been renamed in the Visions Administrator > Site Settings.
string	SpecialService	("Special Service", dtString, Maximum Length 25)
		From a lookup list, select the type of special service for a vessel. Examples: Direct Fired, Lethal, Low Temperature, etc.

Foreign References	
Table Name	Call Value
StatHdr	rfEquipment

EquipDL (rfEquipDL)

Equipment Datalogger ID: the static information associated with equipment types derived from the Boiler and Furnace supplied types to identify a unique identifier that is used in combination with the Section, Elevation and Tube to generate a unique name for each tube that will be surveyed using VisLogger.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	DLShortName	("Equipment DL ID", dtString, Maximum Length 1)
		The number of shell courses for a vessel. Enter a unique value for each piece of equipment as part of the TTL datalogger identifier. Limitation: The value must be unique within the site.

Foreign References	
Table Name	Call Value
StatHdr	rfEquipment

PL_LocationRef (rfPL_LocationRef)

UTM Reference: the UTM (universal transverse Mercator) information converted from earlier versions associated with equipment types derived from the Pipeline Segment, Wellhead, and Pipeline Custom supplied types.

Note: This represents the older pipeline reference information that would have been converted from version 4. These fields are set to read-only by default on an installation, but may be reconfigured by your Visions Administrator.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
decimal?	ContToRefDist	("Control To Ref Distance", dtDouble, Read-Only Field, Unit-Based Field)
		Enter a description of the control or reference point.
string	CPRefDesc	("Control or Ref Point Desc", dtString, Read-Only Field, Maximum Length 25)
		Enter a description of the control or reference point.
string	GPSAccuracy	("Accuracy", dtString, Read-Only Field, Maximum Length 25)
		The accuracy to the GPS coordinate.
string	GPSEasting	("Easting/Longitude", dtString, Read-Only Field, Maximum Length 25)
		The lattitude or GPS easting coordinate.
string	GPSElevation	("Elevation", dtString, Read-Only Field, Maximum Length 25)
		The GPS elevation coordinate.
string	GPSNorthing	("Northing/Latitude", dtString, Read-Only Field, Maximum Length 25)
		The longitude or GPS northing coordinate.
string	GPSZone	("Zone", dtString, Read-Only Field, Maximum Length 25)
		From a lookup list, select the GPS coordinate zone.
string	RefChainage	("Reference Chainage", dtString, Read-Only Field, Maximum Length 15)
		The chainage for the reference of the wellhead or pipeline segment.
string	ReferenceType	("Reference Type", dtString, Read-Only Field, Maximum Length 15)
		From a lookup list, select the type of location reference. Example: Aboveground marker, Girth weld, Valve, etc.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

RE_Couplings (rfRE_Couplings)

Couplings: the coupling information associated with equipment types derived from the Centrifugal Pump, GP Steam Turbine, and Motor supplied types.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)

string	CouplingClass	("Coupling Class", dtString, Maximum Length 20)
		From a lookup list, select the class of coupling.
string	CouplingType	("Coupling Type", dtString, Maximum Length 20)
		From a lookup list, select the type of coupling.
decimal?	CouplSize	("Size", dtDouble, Unit-Based Field)
		The size of the coupling.
string	Manufacturer	("Manufacturer", dtString, Maximum Length 50)
		From a lookup list, select the name of the manufacturer for the coupling.
string	ModelNumber	("Model Number", dtString, Maximum Length 35)
		The model number for the coupling.
int?	ServiceFactor	("Service Factor", dtInteger)
		The service factor for the coupling.
decimal?	SpacerLength	("Spacer Length", dtDouble, Unit-Based Field)
		The nominal length for the spacer of the coupling.
decimal?	SpeedRating	("Speed Rating", dtDouble, Unit-Based Field)
		The operating speed rating for the coupling.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

BOOpening (lstBOOpening)

Furnace Openings: the list of furnace openings information associated with equipment types derived from the Boiler supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
long	**OpenCounter**	("Open Counter", Required Field (PK))
		A counter that uniquely identifies each opening for a boiler.
string	**OpenType**	("Opening Name", Required Field (PK), Maximum Length 25)
		From a lookup list, select the name of the opening into the boiler. Usually refers to the fireside location in the boiler. This may be obtained from the Manufacturer drawings.
string	ConnLocation	("Opening Location", Maximum Length 35)
		The location of the opening on the boiler. This information is obtained from the Manufacturer drawings.
decimal?	OpenDimension	("Opening Height", Unit-Based Field)
		The height of a non-circular opening. This can be obtained from the Manufacturer drawings.

string	OpenIdentification	("Opening Identification", Maximum Length 20)
		From a lookup list, select an identification number or descriptive name for the opening. This may be obtained from the Manufacturer drawings.
decimal?	OpenSize	("Opening Width", Unit-Based Field)
		The width of a non-circular opening or the diameter of a circular opening. This can be obtained from the Manufacturer drawings.
string	OutletType	("Opening Type", Maximum Length 20)
		From a lookup list, select the type of opening. Examples: Flat flange, Insulated quick opening, Hinged door. This may be obtained from the Manufacturer drawings.
string	RefractoryType	("Refractory Type", Maximum Length 20)
		From a lookup list, select the type of refractory or insulation applied to the inside of the door. This may be obtained from the Manufacturer drawings.
decimal?	Thickness	("Thick", Unit-Based Field)
		The thickness of any applied refractory or insulation inside on the inside door surface. This can be obtained from the Manufacturer drawings.

Foreign References	
Table Name	**Call Value**
BoilerStatic	rfEquipment

CoatLin (lstCoatLin)

Coatings: the list of coatings, linings, and cladding information associated a piece of equipment.

<table>
<tr><th>Data Type</th><th>Field</th><th>Field Information (GUI Reference, GUI Data Type, Size, Purpose)</th></tr>
<tr><td rowspan="2">int</td><td rowspan="2"><b>ItemCounter</b></td><td>("Coating Counter", dtInteger, Required Field (PK))</td></tr>
<tr><td>A system counter the uniquely identifies each coating for equipment.</td></tr>
<tr><td rowspan="2">string</td><td rowspan="2">CoatingSystem</td><td>("Coating System", dtString, Maximum Length 35)</td></tr>
<tr><td>From a lookup list, select where the coating system for the equipment, section or nozzle.</td></tr>
<tr><td rowspan="2">string</td><td rowspan="2">CoatLinClad</td><td>("Coating Class", dtString, Maximum Length 10)</td></tr>
<tr><td>From a system lookup list, select the type of protective coating. Choices are:
<table>
<tr><th></th><th>Definition</th></tr>
<tr><td>Coating</td><td>indicates that the information for the coating represents a coating</td></tr>
<tr><td>Lining</td><td>indicates that the information for the coating represents a liner</td></tr>
<tr><td>Cladding</td><td>indicates that the information for the coating represents a cladding</td></tr>
</table></td></tr>
<tr><td rowspan="2">string</td><td rowspan="2">CoatType</td><td>("Coating Type", dtString, Maximum Length 20)</td></tr>
<tr><td>From a lookup list, select the type of coating applied. Examples: Epoxy, Paint, Primer. This may be obtained from the manufacturer construction drawings or the engineering specs.</td></tr>
</table>

string	Comments	("Comments", dtString, Maximum Length 255)
		Enter comments for each coating/lining or cladding record.
string	HowApplied	("How Applied", dtString, Maximum Length 20)
		From a lookup list, select how the coating was applied to the equipment, section or nozzle.
string	Installer	("Installer", dtString, Maximum Length 35)
		From a lookup list, select the name of the organization who installed the applied lining, cladding or protective coating.
string	Manufacturer	("Manufacturer", dtString, Maximum Length 50)
		From a lookup list, select the name of the manufacturer for the coating.
string	Material	("Material", dtString, Maximum Length 25)
		From a lookup list, select the type of material of the applied coating, lining or cladding. This may be obtained from the manufacturer construction drawings or the engineering specs.
decimal?	MinReqdThick	("Min Required Thick", dtDouble, Unit-Based Field)
		The minimum required thickness of the coating. This can be obtained from the design specifications, installers reports or manufacturer drawings.
string	SpecificationNo	("Specification No", dtString, Maximum Length 20)
		From a lookup list, select the specification number for the coating.
decimal?	Thickness	("Thick", dtDouble, Unit-Based Field)
		The actual thickness of the coating material applied. This can be obtained from the manufacturer construction drawings or the engineering specs.
string	WhereApplied	("Where Applied", dtString, Maximum Length 20)
		From a lookup list, select where the coating was applied to the equipment, section or nozzle. This may be obtained from the manufacturer construction drawings or the engineering specs.
int?	YearApplied	("Year Applied", dtInteger)
		The year the coating was applied to the equipment. This can be obtained from the manufacturer construction drawings.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment
GL_LineDetail_V	lstGL_LineDetail_V
GL_PLSegmentLD_V	lstGL_PLSegmentLD_V
Nozzle	lstNozzle
SectionDesign	lstSectionDesign

CTNozzles (lstCTNozzles)

Spray Nozzles: the list of spray nozzle information associated with equipment types derived from the Cooling Tower supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**NozzleID**	("Nozzle ID", dtString, , Required Field (PK), Maximum Length 15)
		The identification number for each spray nozzle in the cooling tower. This can be obtained from the Data Report or the manufacturer construction drawings or sequential nozzle numbering.
string	Manufacturer	("Manufacturer", dtString, Maximum Length 50)
		From a lookup list, select the name of the manufacturer. This may be obtained from the ASME Data Report or stamping.
string	Material	("Material", dtString, Maximum Length 15)
		From a lookup list, select the type of material of the cooling tower nozzles.
string	ModelNumber	("Model Number", dtString, Maximum Length 25)
		The model number of the equipment. This can be obtained from the ASME Data Report or stamping.
decimal?	NozzleSize	("Nozzle Size", dtDouble, Unit-Based Field)
		The nominal diameter of the spray nozzle. This can be obtained from the Data Report or the manufacturer construction drawings.
string	NozzleType	("Nozzle Type", dtString, Maximum Length 15)
		From a lookup list, select the type of spray nozzle.
int?	Quantity	("Quantity", dtInteger)
		The number of spray nozzles of the same type and size within the cooling tower.

Foreign References	
Table Name	**Call Value**
CTStatic	rfEquipment

Elevations (lstElevations)

Elevations: the list of elevation information associated with equipment types derived from the Boiler or Furnace supplied types, used with the Equipment DL ID, Section DL ID and Tube number to create a unique name for each tube within an asset.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**ElevationID**	("Elevation ID", dtString, Required Field (PK), Maximum Length 35)
		Enter a unique identifier for each tube trending elevation.
string	Description	("Elevation Description", dtString, Maximum Length 100)
		Enter a description for the elevation.
string	DLShortName	("Elevation DL ID", dtString, Fixed Length 4)

		Enter a unique value for each elevation which is used as part of the TTL datalogger identifier. Note: The value must use all 4 characters, it is recommended that you pick a prefix as a filler when the value is shorter than 4 characters.
int?	DLSortOrder	("DL Sort Order", dtInteger)
		Enter a numeric value for the datalogger sort order for the elevation.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

EquipRegAuth (lstEquipRegAuth)

Regulatory Authorities: the list of regulatory authority information associated with a piece of equipment.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**RegulatoryAuth**	("Regulatory Authority", dtString, Required Field (PK), Maximum Length 15)
		From a lookup list, select the regulatory authorities that have jurisdiction for the equipment. Examples: ABSA, BC Boilers Branch, ERCB.
string	Description	("Description", dtString, Maximum Length 200)
		The description or explanation for the regulatory authority for the equipment.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

FilterMedia (lstFilterMedia)

Filter Media: the list of media associated with a piece of equipment based on the equipment types derived from the Filter supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**MediaID**	("Media ID", dtString, Required Field (PK), Maximum Length 25)
		Enter a unique identifier for each filter media identified for a filter.
DateTime?	DateInstalled	("Date Installed", dtDate)
		The date the filter media was installed in a filter.
string	Description	("Description", dtString, Maximum Length 150)
		Describe the filter media installed in a filter.
string	Manufacturer	("Manufacturer", dtString, Maximum Length 50)
		From a lookup list, select the name of the manfacturer for the filter media in a filter.
string	Material	("Material", dtString, Maximum Length 15)

		From a lookup list, select the type of material that the filter media is constructed from in a filter.
string	MediaType	("Media Type", dtString, Maximum Length 25)
		From a lookup list, select the type of filter media in a filter.
int?	Quantity	("Quantity", dtInteger)
		The quantity of the filter media in a filter.
decimal?	Thickness	("Thick", dtDouble, Unit-Based Field)
		The thickness of the filter media in a filter.
decimal?	VolWtQty	("Volume", dtDouble, Unit-Based Field)
		The volume of the filter media in a filter.
decimal?	Weight	("Weight", dtDouble, Unit-Based Field)
		The weight of the filter media in a filter.

Foreign References	
Table Name	**Call Value**
FilterStat	rfEquipment

FTStays (lstFTStays)

Stays: the list of stays associated with a piece of equipment based on the equipment types derived from the Firetube Boiler supplied type.

Data Type	**Field**	**Field Information (GUI Reference, GUI Data Type, Size, Purpose)**
long	**StayNum**	("Stay Number", Required Field (PK))
		Enter an identifying stay number for the firetube.
decimal?	DistTubeToShell	("Tube to Shell Dist", dtDouble, Unit-Based Field)
		The distance from the tube to shell for the stay.
string	Location	("Location", dtString, Maximum Length 35)
		From a lookup list, select the location of the stays for the firetube boiler.
decimal?	MAWP	("MAWP", dtDouble, Unit-Based Field)
		The Maximum Allowable Working Pressure of the firetube boiler stays. This would be obtained from the ASME Data Report or the equipment nameplate.
decimal?	MaxPitch	("Max Pitch", dtDouble, Unit-Based Field)
		The maximum pitch for the firetube stay.
long?	NumberStays	("Num of Stays", dtDouble)
		The number of similar stays for the firetube.
decimal?	StaySize	("Stay Size", dtDouble, Unit-Based Field)
		The size of the stay for the firetube.

string	StayType	("Stay Type", dtString, Maximum Length 20)
		From a lookup list, select the type of stay for the firetube boiler.

Foreign References	
Table Name	**Call Value**
FTBoilStat	rfEquipment

FuelTypes (lstFuelTypes)

Fuel Types: the list of types of fuel associated with a piece of equipment based on the equipment types derived from the Boiler, FireTube Boiler, and Furnace supplied types.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**FuelType**	("Fuel Type", dtString, Required Field (PK), Maximum Length 20)
		From a lookup list, select the type of fuel for the firetube boiler.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

GL_CoatLin_V (lstGL_CoatLin_V)

Coatings w/Coordinates: the list of coatings, lining or cladding and their geolocation coordinate information the list of coatings, linings, and cladding information associated a piece of equipment based on the equipment types derived from the Pipe and Pipeline Segment supplied types.

<table>
<tr><th>Data Type</th><th>Field</th><th>Field Information (GUI Reference, GUI Data Type, Size, Purpose)</th></tr>
<tr><td rowspan="2">int</td><td rowspan="2"><b>ItemCounter</b></td><td>("Coating Counter", dtInteger, Required Field (PK))</td></tr>
<tr><td>A system counter the uniquely identifies each coating for equipment.</td></tr>
<tr><td rowspan="2">string</td><td rowspan="2">CoatingSystem</td><td>("Coating System", dtString, Maximum Length 35)</td></tr>
<tr><td>From a lookup list, select where the coating system for the equipment, section or nozzle.</td></tr>
<tr><td rowspan="2">string</td><td rowspan="2">CoatLinClad</td><td>("Coating Class", dtString, Maximum Length 10)</td></tr>
<tr><td>From a system lookup list, select the type of protective coating. Choices are:
<table>
<tr><th></th><th>Definition</th></tr>
<tr><td>Coating</td><td>indicates that the information for the coating represents a coating</td></tr>
<tr><td>Lining</td><td>indicates that the information for the coating represents a liner</td></tr>
<tr><td>Cladding</td><td>indicates that the information for the coating represents a cladding</td></tr>
</table></td></tr>
<tr><td>string</td><td>CoatType</td><td>("Coating Type", dtString, Maximum Length 20)</td></tr>
</table>

		From a lookup list, select the type of coating applied. Examples: Epoxy, Paint, Primer. This may be obtained from the manufacturer construction drawings or the engineering specs.
string	Comments	("Comments", dtString, Maximum Length 255) Enter comments for each coating/lining or cladding record.
string	HowApplied	("How Applied", dtString, Maximum Length 20) From a lookup list, select how the coating was applied to the equipment, section or nozzle.
string	Installer	("Installer", dtString, Maximum Length 35) From a lookup list, select the name of the organization who installed the applied lining, cladding or protective coating.
string	Manufacturer	("Manufacturer", dtString, Maximum Length 50) From a lookup list, select the name of the manufacturer for the coating.
string	Material	("Material", dtString, Maximum Length 25) From a lookup list, select the type of material of the applied coating, lining or cladding.
decimal?	MinReqdThick	("Min Required Thick", dtDouble, Unit-Based Field) The minimum required thickness of the coating.
string	SpecificationNo	("Specification No", dtString, Maximum Length 20) From a lookup list, select the specification number for the coating.
decimal?	Thickness	("Thick", dtDouble, Unit-Based Field) The actual thickness of the coating material applied.
string	WhereApplied	("Where Applied", dtString, Maximum Length 20) From a lookup list, select where the coating was applied to the equipment, section or nozzle.
int?	YearApplied	("Year Applied", dtInteger) The year the coating was applied to the equipment.
bool?	CenterlineFlag	("Use in Centreline", dtInteger) Indicate whether the location coordinates are to be used as part of the dynamic centerline. Options are: 0, N, No, 1, Y, Yes; defaults to N. Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant.
string	Desc_End	("Description - End", dtString, Maximum Length 100) Enter a description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100) Enter a description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20)

		From a system lookup list, select milepost marker reference for the ending milepost distance (chainage); Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant. Note: The lookup list is validated against the Reference Markers Lookup Data.
string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant. Note: The lookup list is validated against the Reference Markers Lookup Data.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field) The ending chainage or linear location. Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field) The starting chainage or linear location. Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger) From a system lookup, select the direction from where the reference is located. Choices are: Left, Right. Perspective is always relative to the increasing direction of chainage. Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger) From a system lookup, select the direction from where the reference is located; Perspective is always relative to the increasing direction of chainage. Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant.
decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located. Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located. Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant.
decimal?	PLPlantFlag	("Pipeline Plant Flag", dtDouble) Identifies whether the plant associated with the asset belongs to a pipeline plant. Limitation: This is a system calculated field, data cannot be populated to it.
decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field) The original ending GPS Longitude (X) location coordinate point.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field)

		The original starting GPS Longitude (X) location coordinate point.
decimal?	Y_EndOrig	("Latitude - End", dtDoubleUnit-Based Field)
		The original ending GPS Latitude (Y) location coordinate point.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Latitude (Y) location coordinate point.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field)
		The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - Start", dtDouble, Unit-Based Field)
		The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment
Nozzle	lstNozzle (rfEquipment, NozzleID)
SectionDesign	lstSectionDesign (rfEquipment, SectionID)

GL_LineDetail_V (lstGL_LineDetail_V)

Line Data w/Coordinates: the list of line numbers and their geolocation coordinate information associated a piece of equipment based on the equipment types derived from the Pipe supplied type.

<table>
<tr><th>Data Type</th><th>Field</th><th>Field Information (GUI Reference, GUI Data Type, Size, Purpose)</th></tr>
<tr><td rowspan="2">string</td><td rowspan="2"><b>LineNum</b></td><td>("Line Number", dtString, Required Field (PK), Maximum Length 35)</td></tr>
<tr><td>A unique identifier for each piping line detail; typically, the specific line or ISO number obtained from the design drawings.</td></tr>
<tr><td rowspan="2">TG17?</td><td rowspan="2">CalcType</td><td>("Minimum Type", dtInteger)</td></tr>
<tr><td>Indicates how the minimum thickness was entered or calculated. Choices are:
<table>
<tr><th></th><th>Definition</th></tr>
<tr><td>NCA</td><td>Indicates that the Nominal - Corrosion Allowance was used to calculate the Minimum Thickness.</td></tr>
<tr><td>Other</td><td>Indicates that an externally calculated, typically an engineering standard, and the Minimum Thickness was user entered.</td></tr>
<tr><td>t-Min Code</td><td>Indicates that the Minimum Thickness was user entered.</td></tr>
<tr><td>t-Min Visions</td><td>Indicates that the Visions t-Min calculator was used to calculate Minimum Thickness.
Limitation: This option is not updateable and only available in the Visions Enterprise Client application when the tMin calculator is used.</td></tr>
</table></td></tr>
<tr><td>decimal?</td><td>Coefficient</td><td>("Coefficient", dtDouble)</td></tr>
</table>

		The y coefficient for the pressure design of piping components; used in the minimum wall thickness and design pressure calculations.
decimal?	CorrAllowance	("Corr Allow", dtDouble, Unit-Based Field)
		The allowed amount for the corrosion of the piping line data material.
decimal?	Diameter	("Nom Diameter", dtDouble, Unit-Based Field)
		Select the NPS or nominal diameter for the piping line detail. The list is built from the Pipe Specifications lookup data.
decimal?	EngStandThick	("Eng Standard Thick", dtDouble, Unit-Based Field)
		The engineering standard thickness for the line data of a pipe.
string	FlangeRating	("Flange Rating", dtString, Maximum Length 10)
		From a lookup list, select the flange rating for the line for piping.
string	FromEquipment	("From Equipment", dtString, Maximum Length 35)
		If applicable, the number or name of the equipment the piping is running from. This may be obtained from the drawings.
string	HeatTracing	("Heat Tracing", dtString, Maximum Length 25)
		From a lookup list, select the type of heat tracing for each line of the piping.
string	InsulationCode	("Insulation Code", dtString, Maximum Length 20)
		From a lookup list, select the insulation code. Examples: H for Hot, C for Cold, PP for Personnel Protection, NB for Noise Mitigation.
decimal?	InsulationThick	("Insulation Thick", dtDoubleUnit-Based Field)
		The thickness of the insulation on the piping line detail.
string	InsulationType	("Insulation Type", dtString, Maximum Length 20)
		From a lookup list, select the type of insulation for each line of the piping.
int?	ItemCounter	("Coating Counter", dtInteger, FK (lstCoatLin))
		Select the coating counter that uniquely links the coating to the line record. Note: The list is sourced from the Coatings (CoatLin) and the value must exist before linking to a line number.
string	LineClass	("Line Class", dtString, Maximum Length 15)
		From a lookup list, select the line classification for each line of the piping.
string	MaterialID	("Material ID", dtString, Maximum Length 15)
		A code determined by the User. Usually indicates the basic material and the year of the Code of reference. For example, A105/98 would be for SA-105 material from the 1998 ASME Codes. Note: The values are sourced from the Material Code Lookup Data and the value must exist before updating a line number.
decimal?	MAWP	("MAWP", dtDouble, Unit-Based Field)
		The Maximum Allowable Working Pressure of the piping line data. This would be obtained from the ASME Data Report or the equipment nameplate.
decimal?	MDWT	("MDWT", dtDouble, Unit-Based Field)

		The Maximum Design Working Temperature. This can be obtained from the Piping Data Report, engineering specifications or drawings.
decimal?	MinDesignThick	("Min Design Thick", dtDouble, Unit-Based Field)
		A user entered value for the ASME Code calculated minimum thickness for the piping line detail.
decimal?	MinReqThickness	("Min Req Thick", dtDouble, Unit-Based Field)
		A user entered value which is the sum of the minimum design thickness plus corrosion allowance. This is the minimum thickness to be ordered for new replacement piping.
decimal?	NomThick	("Nominal", dtDouble, Unit-Based Field)
		The actual new/given thickness of the piping segment. This information is either calculated automatically by Visions if the diameter and schedule are entered or can be obtained from the design specifications or drawings.
decimal?	OD	("OD", dtDouble, Unit-Based Field)
		The outer diameter for the line detail of the pipe; when the nominal diameter is selected, the outside diameter will be defaulted.
decimal?	OpPress	("Operating Press", dtDouble, Unit-Based Field)
		The normal operating pressure of the pipe. This may be obtained from Operations.
decimal?	OpTemp	("Operating Temp", dtDouble, Unit-Based Field)
		The normal operating temperature of the pipe. This may be obtained from Operations.
decimal?	OrigTestPress	("Orig Test Press", dtDouble, Unit-Based Field)
		The original test pressure for the piping line detail.
string	PaintCoat	("Painted/Coated", dtString, Maximum Length 20)
		From a lookup list, select the type of paint coating for each line of the piping.
bool?	PWHT	("PWHT", dtInteger)
		Indicate whether all, part or none of the piping was stress relieved during fabrication. Choices are: 0, N, No, 1, Y, Yes. This may be obtained from the Piping Data Report, engineering specifications or drawings.
decimal?	QualityFactor	("Quality Factor", dtDouble)
		The material quality factor for the piping line detail, used in the minimum wall thickness and design pressure calculations.
string	Radiography	("Radiography", dtString, Maximum Length 15)
		From a lookup list, select the type of radiography for each line of the piping.
string	Reason	("PWHT Reason", dtString, Maximum Length 25)
		From a lookup list, select the reason for the post-weld heat treatment for each line of the piping.
string	Schedule	("Schedule", dtString, Maximum Length 15)
		Select the schedule for the line detail. The list is built from the Pipe Specification lookup data based on the selected NPS.

string	Spec	("Material", dtString, Maximum Length 15) From a lookup list, select the type of material for each line of the piping.
string	SpoolDrawingNum	("Spool Drawing No", dtString, Maximum Length 250) The spool drawing number for the piping line detail.
string	TestType	("Orig Test Type", dtString, Maximum Length 15) From a lookup list, select the type of original test for each line of the piping.
string	ToEquipment	("To Equipment", dtString, Maximum Length 35) If applicable, the number or name of the equipment the piping is running to. This may be obtained from the drawings.
bool?	CenterlineFlag	("Use in Centreline", dtInteger) Indicate whether the location coordinates are to be used as part of the dynamic centerline. Options are: 0, N, No, 1, Y, Yes; defaults to N. Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant.
string	Desc_End	("Description - End", dtString, Maximum Length 100) Enter a description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100) Enter a description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the ending milepost distance (chainage). Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant. Note: The lookup list is validated against the Reference Markers Lookup Data.
string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant. Note: The lookup list is validated against the Reference Markers Lookup Data.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field) The ending chainage or linear location. Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field) The starting chainage or linear location. Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger)

		From a system lookup, select the direction from where the reference is located. Choices are: Left, Right. Perspective is always relative to the increasing direction of chainage. Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger)
		From a system lookup, select the direction from where the reference is located. Choices are: Left, Right. Perspective is always relative to the increasing direction of chainage. Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant.
decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located. Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located. Limitation: Only updatable when the PL module is active and the owning asset belongs to a pipeline plant.
decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field)
		The original ending GPS Longitude (X) location coordinate point.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Longitude (X) location coordinate point.
decimal?	Y_EndOrig	("Latitude - End", dtDouble, Unit-Based Field)
		The original ending GPS Latitude (Y) location coordinate point.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Latitude (Y) location coordinate point.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field)
		The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - Start", dtDouble, Unit-Based Field)
		The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign References	
Table Name	**Call Value**
PipeStat	rfEquipment
CoatLin	lstCoatLin (rfEquipment, ItemCounter)

GL_PLBends_V (lstGLPLBends_V)

PL Bends w/Coordinates: the list of bends and their geolocation coordinate information associated a piece of equipment based on the equipment types derived from the Pipeline Segment supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**BendID**	("Bend ID", dtString, Required Field (PK), Maximum Length 15)
		A unique identifier for each bend of the pipeline segment.
decimal?	AxialDist	("Axial Distance", dtDouble, Unit-Based Field)
		The axial distance for the pipeline segment bend.
decimal?	BendDegree	("Bend Degree", dtDouble)
		The degree of the bend for the pipeline segment.
decimal?	BendRadius	("Radius", dtDouble, Unit-Based Field)
		The radius of the bend for a pipeline segment.
string	BendType	("Type", dtString, Maximum Length 15)
		From a lookup list, select the type of bend for a pipeline segment.
bool?	CenterlineFlag	("Use in Centreline", dtInteger)
		Indicate whether the location coordinates are to be used as part of the dynamic centerline. Options are: 0, N, No, 1, Y, Yes; defaults to N.
string	Desc_End	("Description - End", dtString, Maximum Length 100)
		Enter a description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100)
		Enter a description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20)
		From a system lookup list, select milepost marker reference for the ending milepost distance (chainage). Note: The lookup list is validated against the Reference Markers Lookup Data.
string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20)
		From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). Note: The lookup list is validated against the Reference Markers Lookup Data.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field)
		The ending chainage or linear location.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field)
		The starting chainage or linear location.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger)

		From a system lookup, select the direction from where the reference is located. Choices are: Left, Right. Perspective is always relative to the increasing direction of chainage.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger)
		From a system lookup, select the direction from where the reference is located. Choices are Left, Right. Perspective is always relative to the increasing direction of chainage.
decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located.
decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field)
		The original ending GPS Longitude (X) location coordinate point.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Longitude (X) location coordinate point.
decimal?	Y_EndOrig	("Latitude - End", dtDoubleUnit-Based Field)
		The original ending GPS Latitude (Y) location coordinate point.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Latitude (Y) location coordinate point.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field)
		The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - Start", dtDouble, Unit-Based Field)
		The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign References	
Table Name	**Call Value**
PL_SegmentStat	rfEquipment

GL_PLControls_V (lstGL_PLControls_V)

PL Controls w/Coordinates: the list of controls and their geolocation coordinate information associated a piece of equipment based on the equipment types derived from the Pipeline Segment supplied type.

Data Type	**Field**	**Field Information (GUI Reference, GUI Data Type, Size, Purpose)**
string	**ControlID**	("Control ID", dtString, Required Field (PK), Maximum Length 15)
		A unique identifier for each control on the pipeline segment.
string	ControlType	("Control Type", dtString, Maximum Length 15)

		From a lookup list, select the type of control for the pipeline segment; such as buoyancy controls, exposed or bare sections, etc.
decimal?	EndDist	("End Distance", dtDouble, Unit-Based Field)
		The ending axial distance for the control on the pipeline segment.
string	Reference	("Reference", dtString, Maximum Length 25)
		The reference for the control on the pipeline segment.
decimal?	StartDist	("Start Distance", dtDouble, Unit-Based Field)
		The starting axial distance for the control on the pipeline segment.
bool?	CenterlineFlag	("Use in Centreline", dtInteger)
		Indicate whether the location coordinates are to be used as part of the dynamic centerline. Options are: 0, N, No, 1, Y, Yes; defaults to N.
string	Desc_End	("Description - End", dtString, Maximum Length 100)
		Enter a description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100)
		Enter a description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20)
		From a system lookup list, select milepost marker reference for the ending milepost distance (chainage). Note: The lookup list is validated against the Reference Markers Lookup Data.
string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20)
		From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). Note: The lookup list is validated against the Reference Markers Lookup Data.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field)
		The ending chainage or linear location; only visible when the PL module is active.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field)
		The starting chainage or linear location; only visible when the PL module is active.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger)
		From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger)
		From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage.
decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located.

decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field)
		The original ending GPS Longitude (X) location coordinate point.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Longitude (X) location coordinate point.
decimal?	Y_EndOrig	("Latitude - End", dtDoubleUnit-Based Field)
		The original ending GPS Latitude (Y) location coordinate point.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Latitude (Y) location coordinate point.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field)
		The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - Start", dtDouble, Unit-Based Field)
		The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign References	
Table Name	**Call Value**
PL_SegmentStat	rfEquipment

GL_PLSegCrossings_V (lstGL_PLSegCrossings_V)

PL Crossings w/Coordinates: the list of crossings and their geolocation coordinate information associated a piece of equipment based on the equipment types derived from the Pipeline Segment supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**CrossingID**	("Crossing ID", dtString, Required Field (PK), Maximum Length 15)
		Enter a unique identifier for each crossing of a pipeline segment.
string	AnchorDevice	("Anchor Device", dtString, Maximum Length 25)
		From a lookup list, select the type of anchor device that is used at the crossing of a pipeline segment.
long?	Angle	("Angle", dtDouble, Unit-Based Field)
		The angle or degrees of the crossing.
string	ConstructionType	("Construction Type", dtString, Maximum Length 15)
		From a lookup list, select the type of construction for each crossing of a pipeline segment. Examples: Cased, directional drill, etc.
string	CrossingRefID	("Crossing Reference ID", dtString, Maximum Length 25)
		The reference identification for each crossing of a pipeline segment.
string	CrossingType	("Crossing Type", dtString, Maximum Length 20)

		From a lookup list, select the type of crossing for a pipeline segment.
DateTime?	DateDone	("Date Bonding Done", dtDate)
		The date the bonding was done for each crossing of a pipeline segment.
decimal?	Depth	("Depth", dtDouble, Unit-Based Field)
		The depth of the crossing for a pipeline segment.
string	Description	("Description", dtString, Maximum Length 35)
		Describe the crossing for a pipeline segment.
string	Direction	("Direction", dtString, Maximum Length 30)
		Describe the direction that the crossing pipeline is going.
bool?	HasCP	("Has CP", dtInteger)
		Indicate whether the crossing has cathodic protection. Choices are: 0, N, No, 1, Y, Yes.
string	Location	("Location", dtString, Maximum Length 35)
		The location for each crossing of a pipeline segment.
string	Orientation	("Orientation", dtString, Maximum Length 15)
		From a lookup list, select the orientation of the crossing for a pipeline segment.
string	PhysicalType	("Physical Type", dtString, Maximum Length 15)
		From a lookup list, select the type of physical crossing. Examples: Railway, Highway, etc.
bool?	PLSegBonded	("Pipeline Segment Bonded", dtInteger)
		Identify whether the pipeline segment is bonded. Choices are: 0, N, No, 1, Y, Yes.
decimal?	PLSeparation	("Pipeline Separation", dtDouble, Unit-Based Field)
		The separation distance between two pipeline segments for each crossing.
string	SuspDevice	("Suspension Device", dtString, Maximum Length 25)
		From a lookup list, select the type of suspension device for each crossing for a pipeline segment.
bool?	CenterlineFlag	("Use in Centreline", dtInteger)
		Indicate whether the location coordinates are to be used as part of the dynamic centerline. Options are: 0, N, No, 1, Y, Yes; defaults to N.
string	Desc_End	("Description - End", dtString, Maximum Length 100)
		Enter a description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100)
		Enter a description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20)

		From a system lookup list, select milepost marker reference for the ending milepost distance (chainage). Note: The lookup list is validated against the Reference Markers Lookup Data.
string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20)
		From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). Note: The lookup list is validated against the Reference Markers Lookup Data.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field)
		The ending chainage or linear location; only visible when the PL module is active.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field)
		The starting chainage or linear location; only visible when the PL module is active.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger)
		From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger)
		From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage.
decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located.
decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field)
		The original ending GPS Longitude (X) location coordinate point.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Longitude (X) location coordinate point.
decimal?	Y_EndOrig	("Latitude - End", dtDoubleUnit-Based Field)
		The original ending GPS Latitude (Y) location coordinate point.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Latitude (Y) location coordinate point.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field)
		The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - Start", dtDouble, Unit-Based Field)
		The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign References

Table Name	Call Value
PL_SegmentStat	rfEquipment

GL_PLSegmentLD_V (lstGL_PLSegmentLD_V)

PL Joint/Line Data w/Coordinates: the list of joint or line numbers and their geolocation coordinate information associated a piece of equipment based on the equipment types derived from the Pipeline Segment supplied type.

<table>
<tr><th>Data Type</th><th>Field</th><th>Field Information (GUI Reference, GUI Data Type, Size, Purpose)</th></tr>
<tr><td rowspan="2">string</td><td rowspan="2"><b>LineID</b></td><td>("Line ID", dtString, Required Field (PK), Maximum Length 35)</td></tr>
<tr><td>Enter a unique identifier for each line or joint for a pipeline segment.</td></tr>
<tr><td rowspan="2">TG17?</td><td rowspan="2">CalcType</td><td>("Minimum Type", dtInteger)</td></tr>
<tr><td>Indicates how the minimum thickness was entered or calculated. Choices are:
<table>
<tr><th></th><th>Definition</th></tr>
<tr><td>NCA</td><td>Indicates that the Nominal - Corrosion Allowance was used to calculate the Minimum Thickness.</td></tr>
<tr><td>Other</td><td>Indicates that an externally calculated, typically an engineering standard, was used to calculate the Minimum Thickness.</td></tr>
<tr><td>t-Min Code</td><td>Indicates that the Minimum Thickness was user entered.</td></tr>
<tr><td>t-Min Visions</td><td>Indicates that the Visions t-Min calculator was used to calculate Minimum Thickness.
Limitation: This option is not updateable and only available in the Visions Enterprise Client application when the tMin calculator is used.</td></tr>
</table></td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">CorrAllow</td><td>("Corr Allow", dtDouble, Unit-Based Field)</td></tr>
<tr><td>The corrosion allowance for each line size / class for a pipeline segment.</td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">DesignFactor</td><td>("Design Factor", dtDouble)</td></tr>
<tr><td>The design factor for a pipeline segment. (Taken from CSA-Z-662 section 4.3.3.2 and used to calculate the design pressure.)</td></tr>
<tr><td rowspan="2">int?</td><td rowspan="2">DesignLife</td><td>("Design Life", dtInteger, Unit-Based Field)</td></tr>
<tr><td>The number of years the joint/line is designed for.</td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">EngStandThick</td><td>("Eng Standard Thick", dtDouble, Unit-Based Field)</td></tr>
<tr><td>The engineering standard thickness for the joint or line data of a pipeline segment.</td></tr>
<tr><td rowspan="2">string</td><td rowspan="2">FlangeRating</td><td>("Flange Rating", dtString, Maximum Length 10)</td></tr>
<tr><td>From a lookup list, select the ANSI rating of the joint/line flange for a pipeline segment.</td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">Hydrotest</td><td>("Hydrotest", dtDouble, Unit-Based Field)</td></tr>
<tr><td>The hydrotest pressure for a pipeline segment.</td></tr>
<tr><td>bool?</td><td>ILIPerform</td><td>("ILI Performable", dtInteger)</td></tr>
</table>

		Identify whether inline inspection is performable for a pipeline segment. Choices are: 0, N, No, 1, Y, Yes.
int?	ItemCounter	("Coating Counter", dtInteger, FK (lstGL_CoatLin_V))
		Select the coating counter that uniquely links the coating to the joint/line record. Note: The list is sourced from the Coatings (GL_CoatLin_V) and the value must exist before linking to a line ID.
string	JointCode	("Joint Code", dtString, Maximum Length 15)
		From a lookup list, select the code for the type of joint for each line size / class for a pipeline segment. (Taken from EUB Guide 56, table 6.11)
decimal?	JointFactor	("Joint Factor", dtDouble)
		The joint factor for a pipeline segment. (Taken from CSA-Z-662 section 4.3.3.4 and used to calculate the design pressure.)
decimal?	LDLength	("Length", dtDouble, Unit-Based Field)
		The length for each line size / class for a pipeline segment.
decimal?	LineSize	("Line Size", dtDouble, Unit-Based Field)
		Select the size of the pipe for each line size / class for a pipeline segment. Note: The list is validated against the Pipe Specification lookup data.
decimal?	LocationFactor	("Location Factor", dtDouble)
		The location factor for a pipeline segment.
string	LocationFrom	("Location From", dtString, Maximum Length 35)
		The from location for each line size / class for a pipeline segment.
string	MaterialID	("Material ID", dtString, Maximum Length 15)
		A code determined by the User. Usually indicates the basic material and the year of the Code of reference. For example, A105/98 would be for SA-105 material from the 1998 ASME Codes. Note: The values are sourced from the Material Code Lookup Data and the value must exist before updating a line number.
string	MaterialCode	("Material Code", dtString, Maximum Length 5)
		From a lookup list, select the material code for each line size / class for a pipeline segment. Examples: A, G, F, etc. (Taken from EUB Guide 56, table 6.6).
decimal?	MinThick	("Min Thick", dtDouble, Unit-Based Field)
		The minimum thickness for each line size / class for a pipeline segment.
decimal?	NomThick	("Nominal", dtDouble, Unit-Based Field)
		The nominal thickness for each line size / class for a pipeline segment.
decimal?	OD	("OD", dtDouble, Unit-Based Field)
		The outer diameter for each line size / class for a pipeline segment.
bool?	PiggingPerform	("Pigging Performable", dtInteger)
		Identify whether pigging is performable for the pipeline segment. Choices are: 0, N, No, 1, Y, Yes.
string	Schedule	("Schedule", dtString, Maximum Length 15)

		Select the schedule for the line detail. The list is built from the Pipe Specification lookup data based on the selected NPS.
string	SeamWeldType	("Seam Weld Type", dtString, Maximum Length 15)
		From a lookup list, select the type of seam weld for each line size / class for a pipeline segment. Examples: Butt-weld, Lapweld, Seamless, etc.
decimal?	SMYS	("SMYS", dtDouble, Unit-Based Field)
		The specified minimum yield strength (SMYS) for each line size / class for a pipeline segment.
decimal?	StressLevel	("Stress Level", dtDouble)
		The stress level percentage for each line size / class for a pipeline segment. Stress level is defined as the stress in the wall of a pipe that is produced by the pressure of fluids in the pipeline.
decimal?	TemperatureFactor	("Temp Factor", dtDouble)
		The temperature factor for the pipeline segment.
bool?	CenterlineFlag	("Use in Centreline", dtInteger)
		Indicate whether the location coordinates are to be used as part of the dynamic centerline. Options are: 0, N, No, 1, Y, Yes; defaults to N.
string	Desc_End	("Description - End", dtString, Maximum Length 100)
		Enter a description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100)
		Enter a description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20)
		From a system lookup list, select milepost marker reference for the ending milepost distance (chainage). Note: The lookup list is validated against the Reference Markers Lookup Data.
string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20)
		From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). Note: The lookup list is validated against the Reference Markers Lookup Data.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field)
		The ending chainage or linear location.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field)
		The starting chainage or linear location.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger)
		From a system lookup, select the direction from where the reference is located, choices are: Left, Right. Perspective is always relative to the increasing direction of chainage.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger)

		From a system lookup, select the direction from where the reference is located. Choices are: Left, Right. Perspective is always relative to the increasing direction of chainage.
decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located.
decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field) The original ending GPS Longitude (X) location coordinate point.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field) The original starting GPS Longitude (X) location coordinate point.
decimal?	Y_EndOrig	("Latitude - End", dtDoubleUnit-Based Field) The original ending GPS Latitude (Y) location coordinate point.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field) The original starting GPS Latitude (Y) location coordinate point.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field) The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - Start", dtDouble, Unit-Based Field) The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign References	
Table Name	**Call Value**
PL_SegmentStat	rfEquipment
GL_CoatLin_V	lstGL_CoatLin_V (rfEquipment, ItemCounter)

GL_PLSoilInfo_V (lstGL_PLSoilInfo_V)

PL Soil Info w/Coordinates: the list of soil layers and their geolocation coordinate information associated a piece of equipment based on the equipment types derived from the Pipeline Segment supplied type.

Data Type	**Field**	**Field Information (GUI Reference, GUI Data Type, Size, Purpose)**
string	**SoilID**	("Soil Layer ID", dtString, Required Field (PK), Maximum Length 15) Enter a unique identifier for each soil layer for a pipeline segment.
decimal?	AxialDist	("Axial Distance", dtDouble, Unit-Based Field) The axial distance for each soil layer for a pipeline segment.
decimal?	CenterDist	("Centreline Distance", dtDouble, Unit-Based Field)

		The centerline distance for each soil layer for a pipeline segment.
decimal?	CoverDepth	("Cover Depth", dtDouble, Unit-Based Field)
		The depth of cover for each soil layer for a pipeline segment.
string	Deposition	("Soil Deposition", dtString, Maximum Length 15)
		From a lookup list, select the type of soil deposition for a pipeline segment. Examples: Morainal, Lacustrine, Fluvial, etc.
string	Drainage	("Drainage", dtString, Maximum Length 15)
		From a lookup list, select the type of drainage for a pipeline segment. Examples: Well, Very Poor, Imperfect, etc.
DateTime?	ReadingDate	("Reading Date", dtDate)
		The date of the resistivity reading for a pipeline segment.
decimal?	Resistivity	("Resistivity", dtDouble, Unit-Based Field)
		The resistivity measurement for a pipeline segment.
bool?	ResistivityMeasured	("Resistivity Measured", dtInteger)
		Identify whether resistivity is measured for a pipeline segment. Choices are: 0, N, No, 1, Y, Yes.
string	ResistivityMethod	("Resistivity Method", dtString, Maximum Length 15)
		From a lookup list, select the type of method used to measure resistivity for a pipeline segment. Examples: Soil Box, Wenner, etc.
string	StratificationID	("Stratification Type", dtString, Maximum Length 15)
		From a lookup list, select the type of stratification for each soil layer for a pipeline segment. Examples: Mottling, Gleying, Organic, Bedrock, etc.
decimal?	TextureEnd	("End", dtDouble, Unit-Based Field)
		The end of a soil texture for each soil layer for a pipeline segment.
string	TextureID	("Texture ID", dtString, Maximum Length 15)
		From a lookup list, select the identifier for a soil texture for each soil layer for a pipeline segment.
decimal?	TextureStart	("Start", dtDouble, Unit-Based Field)
		The start of a soil texture for each soil layer for a pipeline segment.
bool?	CenterlineFlag	("Use in Centreline", dtInteger)
		Indicate whether the location coordinates are to be used as part of the dynamic centerline. Options are: 0, N, No, 1, Y, Yes; defaults to N.
string	Desc_End	("Description - End", dtString, Maximum Length 100)
		Enter a description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100)
		Enter a description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20)

		From a system lookup list, select milepost marker reference for the ending milepost distance (chainage). Note: The lookup list is validated against the Reference Markers Lookup Data.
string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20)
		From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). Note: The lookup list is validated against the Reference Markers Lookup Data.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field)
		The ending chainage or linear location.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field)
		The starting chainage or linear location.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger)
		From a system lookup, select the direction from where the reference is located, choices are: Left, Right. Perspective is always relative to the increasing direction of chainage.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger)
		From a system lookup, select the direction from where the reference is located. Choices are: Left, Right. Perspective is always relative to the increasing direction of chainage.
decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located.
decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field)
		The original ending GPS Longitude (X) location coordinate point.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Longitude (X) location coordinate point.
decimal?	Y_EndOrig	("Latitude - End", dtDoubleUnit-Based Field)
		The original ending GPS Latitude (Y) location coordinate point.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Latitude (Y) location coordinate point.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field)
		The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - Start", dtDouble, Unit-Based Field)
		The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign References

Table Name	Call Value
PL_SegmentStat	rfEquipment

GL_PLTopography_V (lstGL_PLTopography_V)

PL Topography w/Coordinates: the list of topography and their geolocation coordinate information associated a piece of equipment based on the equipment types derived from the Pipeline Segment supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**TopographyID**	("Topography ID", dtString, Required Field (PK), Maximum Length 15) Enter a unique identifier for each topographic feature of a pipeline segment.
decimal?	AdjacentSlope	("Adjacent Slopes", dtDouble, Unit-Based Field) The average angle of an adjacent slope, in degrees, for a pipeline segment.
string	LandUse	("Land Use", dtString, Maximum Length 25) From a lookup list, select the type of land use for each topographic feature of a pipeline segment. Examples: Urban, Industrial, Grazing, Cultivated, etc.
string	PhysioCategory	("Physio-Category", dtString, Maximum Length 25) From a lookup list, select the physiographic category for each topographic feature of a pipeline segment. Examples: Alpine, Volcanic Areas, Sedimentary Cover, etc.
string	PhysioRegion	("Physiographic Region", dtString, Maximum Length 25) From a lookup list, select the physiographic region for each topographic feature of a pipeline segment. Examples: St. Lawrence Lowlands, Canadian Shield, Arctic, Hudson Platform, etc.
string	SitePosition	("Site Position", dtString, Maximum Length 25) From a lookup list, select the site position for each topographic feature of a pipeline segment. Examples: Level, Depression, Toe, Crest, Mid Slope, etc.
string	Topography	("Topography", dtString, Maximum Length 25) From a lookup list, select the type of topography for each topographic feature of a pipeline segment. Examples: Level, Undulating, Depression, Ridged, etc.
string	VegetativeLand	("Vegetative Landform", dtString, Maximum Length 25) From a lookup list, select the type of vegetative landform for the topography of a pipeline segment. Examples: Grasslands, Boreal, Deciduous, etc.
string	Desc_End	("Description - End", dtString, Maximum Length 100) Enter a description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100) Enter a description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the ending milepost distance (chainage). Note: The lookup list is validated against the Reference Markers Lookup Data.

string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20)
		From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). Note: The lookup list is validated against the Reference Markers Lookup Data.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field)
		The ending chainage or linear location.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field)
		The starting chainage or linear location.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger)
		From a system lookup, select the direction from where the reference is located, choices are: Left, Right. Perspective is always relative to the increasing direction of chainage.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger)
		From a system lookup, select the direction from where the reference is located. Choices are: Left, Right. Perspective is always relative to the increasing direction of chainage.
decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located.
decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field)
		The original ending GPS Longitude (X) location coordinate point.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Longitude (X) location coordinate point.
decimal?	Y_EndOrig	("Latitude - End", dtDoubleUnit-Based Field)
		The original ending GPS Latitude (Y) location coordinate point.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Latitude (Y) location coordinate point.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field)
		The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - Start", dtDouble, Unit-Based Field)
		The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign References	
Table Name	**Call Value**
PL_SegmentStat	rfEquipment

GL_PLWaterDepth_V (lstGL_PLWaterDepth_V)

PL Water Depths w/Coordinates: the list of water depths and their geolocation coordinate information associated a piece of equipment based on the equipment types derived from the Pipeline Segment supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**WaterDepthID**	("Water Depth ID", dtString, Required Field (PK), Maximum Length 15) Enter a unique identifier for each water depth for a pipeline segment.
decimal?	BuriedDepth	("Buried Depth", dtDouble, Unit-Based Field) If the pipeline segment is buried, The depth it is buried.
bool?	BuriedFlag	("Buried", dtInteger) Identify whether the pipeline segment is buried. Choices are: 0, N, No, 1, Y, Yes.
decimal?	WaterDepth	("Water Depth", dtDouble, Unit-Based Field) The water depth for a pipeline segment.
bool?	CenterlineFlag	("Use in Centreline", dtInteger) Indicate whether the location coordinates are to be used as part of the dynamic centerline. Options are: 0, N, No, 1, Y, Yes; defaults to N.
string	Desc_End	("Description - End", dtString, Maximum Length 100) Enter a description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100) Enter a description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the ending milepost distance (chainage). Note: The lookup list is validated against the Reference Markers Lookup Data.
string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). Note: The lookup list is validated against the Reference Markers Lookup Data.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field) The ending chainage or linear location.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field) The starting chainage or linear location.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger) From a system lookup, select the direction from where the reference is located, choices are: Left, Right. Perspective is always relative to the increasing direction of chainage.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger)

		From a system lookup, select the direction from where the reference is located. Choices are: Left, Right. Perspective is always relative to the increasing direction of chainage.
decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located.
decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field)
		The original ending GPS Longitude (X) location coordinate point.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Longitude (X) location coordinate point.
decimal?	Y_EndOrig	("Latitude - End", dtDoubleUnit-Based Field)
		The original ending GPS Latitude (Y) location coordinate point.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Latitude (Y) location coordinate point.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field)
		The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - Start", dtDouble, Unit-Based Field)
		The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign References	
Table Name	**Call Value**
PL_SegmentStat	rfEquipment

HeatSurface (lstHeatSurface)

Heating Surface: the list of heating surface information associated a piece of equipment based on the equipment types derived from the Boiler supplied type.

Data Type	**Field**	**Field Information (GUI Reference, GUI Data Type, Size, Purpose)**
string	**HeatSurfType**	("Boiler Section", dtString, Required Field (PK), Maximum Length 25)
		From a lookup list, select the applicable boiler section for the heating surface of the boiler.
decimal?	Hydrotest	("Hydrotest", dtDouble, Unit-Based Field)
		The hydrostatic test pressure for the boiler heating surface section. This would be obtained from the ASME Data Report.
decimal?	MAWP	("MAWP", dtDouble, Unit-Based Field)
		The Maximum Allowable Working Pressure of the boiler heating surface. This would be obtained from the ASME Data Report or the equipment nameplate.

decimal?	MAWT	("MAWT", dtDouble, Unit-Based Field)
		The Maximum Allowable Working Temperature for the boiler heating surface section. This can be obtained from the ASME Data Report or the boiler nameplate.
decimal?	mHeatSurface	("Heat Surface", dtDouble, Unit-Based Field)
		The registerted heating surface for the boiler section. This would be obtained from the ASME Data Report.

Foreign References	
Table Name	**Call Value**
BoilerStatic	rfEquipment

NotesStat (lstNotesStat)

Notes (Static Data): the list of notes associated on the Static Data for a piece of equipment.

Data Type	**Field**	**Field Information (GUI Reference, GUI Data Type, Size, Purpose)**
DateTime	**NotesDate**	("Notes Date", dtDate, Required Field (PK))
		The date of the additional comments or notes on the equipment.
string	Notes	("Notes", dtString, Maximum Length 4000)
		A descriptive, explanation for any additional comments or notes on the equipment.
string	NotesCategory	("Notes Category", dtString, Maximum Length 25)
		From a lookup list, select a category for the notes.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

Nozzle (lstNozzle)

Nozzles: the list of nozzles associated on the Static Data for a piece of equipment.

Data Type	**Field**	**Field Information (GUI Reference, GUI Data Type, Size, Purpose)**
string	**NozzleID**	("Nozzle Number", dtString, Required Field (PK), Maximum Length 10)
		Enter a unique nozzle number for each nozzle for equipment. Recommendation: user the nozzle identification number from the ASME/API Data Report, the manufacturers construction drawing or sequential nozzle numbering.
TG17?	CalcType	("Minimum Type", dtInteger)
		Indicates how the minimum thickness was entered or calculated. Choices are: Definition NCA Indicates that the Nominal - Corrosion Allowance was used to calculate the Minimum Thickness.

Other	Indicates that an externally calculated, typically an engineering standard, was used to calculate the Minimum Thickness.
t-Min Code	Indicates that the Minimum Thickness was user entered.
t-Min Visions	Indicates that the Visions t-Min calculator was used to calculate Minimum Thickness. Limitation: This option is not updateable and only available in the Visions Enterprise Client application when the tMin calculator is used.

decimal?	DiameterOrSize	("Diameter/Width", dtDouble, Unit-Based Field) Enter or select from a list, the diameter or size of a nozzle. The list is built from the Pipe Specification lookup data. This may be obtained from the ASME/API data report or the manufacturer construction drawing.
decimal?	EngStandThick	("Eng Standard Thick", dtDouble, Unit-Based Field) The engineering standard thickness for the nozzle.
string	FlangeHowAttached	("Flange: How Attached", dtString, Maximum Length 20) From a lookup list, select the method of attaching the flange to the nozzle neck. Examples: Welded, Type 1 Weld. This can be obtained from the ASME/API Data Report or the manufacturers construction drawing.
string	FlgMaterialID	("Material ID", dtString, Maximum Length 15) A code determined by the User. Usually indicates the basic material and the year of the Code of reference. For example, A105/98 would be for SA-105 material from the 1998 ASME Codes. Note: The values are sourced from the Material Code Lookup Data and the value must exist before updating a line number.
string	FlangeType	("Connection Type", dtString, Maximum Length 15) From a lookup list, select the type of flange connection on the nozzle. Examples: RFWN, RFSO, RTJ. This can be obtained from the ASME/API Data Report or the manufacturers construction drawing.
decimal?	Height	("Height", dtDouble, Unit-Based Field) The height for the nozzle. This can be obtained from the ASME/API Data Report or the manufacturers construction drawing.
int?	InstallDate	("Install Year", dtInteger) The year the nozzle was installed.
int?	ItemCounter	("Coating Counter", dtInteger, FK (lstCoatLin or rfGL_CoatLin_V)) Select the coating counter that uniquely links the coating to the joint/line record. Note: The list is sourced from the Coatings (CoatLin) and the value must exist before linking to a nozzle.
string	Location	("Location", dtString, Maximum Length 35) From a lookup list, selec the location of the nozzle on the equipment. Example: East Shell, Roof, etc.
string	NozzFlangeRating	("Connection Rating", dtString, Maximum Length 10)

		From a lookup list, select the ANSI connection rating of the nozzle flange. This can be obtained from the ASME/API Data Report or the manufacturers construction drawing.
string	NozzleHowAttached	("Nozzle: How Attached", dtString, Maximum Length 20) From a lookup list, select the method of attaching the nozzle fo the equipment. Examples: Welded, UW16.1(e). This can be obtained from the ASME/API Data Report or the manufacturers construction drawing.
string	NzMaterialID	("Material ID", dtString, Maximum Length 15) A code determined by the User. Usually indicates the basic material and the year of the Code of reference. For example, A105/98 would be for SA-105 material from the 1998 ASME Codes. Note: The values are sourced from the Material Code Lookup Data and the value must exist before updating a line number.
decimal?	NozzleNeckCA	("Nozzle Neck CA", dtDouble, Unit-Based Field) The corrosion allowance for the nozzle neck. This can be obtained from the ASME/API Data Report or the manufacturers construction drawing.
decimal?	NozzleNeckMin	("Nozzle Neck Min", dtDouble, Unit-Based Field) The minimum thickness for the nozzle neck. This can be obtained from the ASME/API Data Report or the manufacturers construction drawing.
decimal?	NozzleNominal	("Nozzle Neck Nominal", dtDouble, Unit-Based Field) The nozzle neck nominal thickness. If the lookup is used for the diameter and schedule, it will be defaulted from the Pipe Specifications. This can be obtained from the ASME/API Data Report or the manufacturers construction drawing.
string	NozzleShape	("Nozzle Shape", dtString, Maximum Length 15) From a lookup list, select the shape of the nozzle. This may be obtained from the ASME/API Data Report or the manufacturers construction drawing.
string	Purpose	("Purpose", dtString, Maximum Length 20) From a lookup list, select the purpose or name of the nozzle. This can be obtained from the ASME/API Data Report or the manufacturers drawing.
int?	Quantity	("Quantity", dtInteger) The quantity of specific nozzles. Example: level gauges have two similar nozzles. This can be obtained from the ASME/API Data Report or the manufacturers construction drawing.
string	ReinforMaterial	("Reinforcement Material", dtString, Maximum Length 15) From a lookup list, select the type of material for the nozzle reinforcement pad (if any). This can be obtained from the ASME/API Data Report or the manufacturers construction drawing.
decimal?	RepadOD	("Repad OD", dtDouble, Unit-Based Field) The outside diameter of the nozzle reinforcement pad. This can be obtained from the manufacturers construction drawing.
decimal?	RepadThick	("Repad Thick", dtDouble, Unit-Based Field) The thickness of the nozzle reinforcement pad. This can be obtained from the manufacturers construction drawing.

string	Schedule	("Schedule", dtString, Maximum Length 15)
		From a list, select the schedule of the nozzle piping neck. If selected, Visions will automatically default the nominal neck thickness. This can be obtained from the ASME/API Data Report or manufacturers construction drawing.
string	StreamID	("Stream ID", dtString, Maximum Length 20, FK (rfPltExchStreams))
		Select the plate exchanger stream for the nozzle. Note: The list is sourced from the Plate Exchanger Streams (lstPltExchStreams) and the value must exist before linking to a nozzle. Limitation: This field is only available for equipment types derived from the supplied Plate Exchanger.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment
CoatLin	lstCoatLin (rfEquipment, ItemCounter)
GL_CoatLin_V	lstGL_CoatLin_V (rfEquipment, ItemCounter)
PltExchStreams	lstPltExchStreams (rfEquipment, StreamID

PL_Refs (lstPL_Refs)

PL References: the list of reference markers and associated chainages for a piece of equipment based on the equipment types derived from the Pipeline Segment supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**MarkerID**	("Reference ID", dtString, Required Field (PK), Maximum Length 20)
		A unique system identifier to link the location coordinates to the pipeline segment. Note: The lookup list is validated against the Reference Markers Lookup Data.
decimal?	MP_Chainage	("MP (Chainage)", dtDouble, Unit-Based Field)
		The ending chainage or linear location; only visible when the PL module is active.
TG56?	OffsetDir	("Offset Direction", dtInteger)
		From a system lookup, select the direction from where the reference is located. Choices are: Left, Right. Perspective is always relative to the increasing direction of chainage.
decimal?	OffsetDist	("Offset Distance", dtDouble, Unit-Based Field)
		The distance perpendicular to the pipe from where the reference is located.

Foreign References	
Table Name	**Call Value**
PL_SegmentStat	rfEquipment

PL_SegElevations (lstPL_SegElevations)

PL Elevations: the list of elevations associated with a piece of equipment based on the equipment types derived from the Pipeline Segment supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
decimal	**ElevDistance**	("Elevation Distance", dtDouble, Required Field (PK), Unit-Based Field)
		The elevation distance for each elevation for a pipeline segment.
decimal?	Elevation	("Elevation", dtDouble, Unit-Based Field)
		Enter a unique elevation for a pipeline segment.

Foreign References	
Table Name	**Call Value**
PL_SegmentStat	rfEquipment

PltExchStreams (lstPltExchStreams)

Plate Exchanger Streams: the list of streams associated with a piece of equipment based on the equipment types derived from the Plate Exchanger supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**StreamID**	("Stream ID", dtString, Required Field (PK), Maximum Length 20, FK (lstNozzle))
		Enter a unique identifier for each process stream for the plate exchanger.
decimal?	DesignPress	("Design Press", dtDoubleUnit-Based Field)
		The design pressure for the plate exchanger process stream.
decimal?	DesignTemp	("Design Temp", dtDouble, Unit-Based Field)
		The design temperature for the plate exchanger process stream.
bool?	FullVacuum	("Full Vacuum", dtInteger)
		Indicate whether the plate exchanger stream is full vacuum. Choices are: 0, N, No, 1, Y, Yes.
decimal?	MAWP	("MAWP", dtDouble, Unit-Based Field)
		The Maximum Allowable Working Pressure of the plate exhanger stream. This would be obtained from the ASME Data Report or the equipment nameplate.
decimal?	MAWT	("MAWT", dtDouble, Unit-Based Field)
		The Maximum Allowable Working Temperature for the plate exchanger process stream. This would be from the ASME Data report or the equipment nameplate.
decimal?	MDMT	("MDMT", dtDouble, Unit-Based Field)
		The maximum pressure allowed at the Minimum Design Metal Temperature for the plate exchanger process stream. This could be obtained from the ASME Data Report or the equipment nameplate.
decimal?	MDMTPress	("MDMT Press", dtDouble, Unit-Based Field)

		The maximum pressure allowed at the Minimum Design Metal Temperature for the plate exchanger. This could be obtained from the ASME Data Report or the equipment nameplate.
bool?	NonPressure	("Non-Pressure", dtInteger)
		Indicate whether the plate exchanger stream is considered non-pressure. Choices are: 0, N, No, 1, Y, Yes.
string	NPReason	("NP Reason", dtString, Maximum Length 25)
		From a lookup list, select the reason why (vented to atmosphere, etc.) the tubeside is considered non-pressure. This may be obtained from the data report or design specifications.
decimal?	OperPress	("Operating Press", dtDouble, Unit-Based Field)
		The operating pressure for the plate exchanger process stream.
decimal?	OpTemp	("Operating Temp Inlet", dtDouble, Unit-Based Field)
		The inlet operating temperature for the plate exchanger process stream.
decimal?	OpTempOut	("Operating Temp Outlet", dtDouble, Unit-Based Field)
		The outlet operating temperature for the plate exchanger process stream.
decimal?	OrigTestPress	("Orig Test Press", dtDouble, Unit-Based Field)
		The original test pressure for the plate exchanger process stream.
bool?	SourService	("Sour Service", dtInteger)
		Indicate whether the plate exchanger process stream is in sour service. Choices are: 0, N, No, 1, Y, Yes.
string	TestType	("Orig Test Type", dtString, Maximum Length 15)
		From a lookup list, select the type of testing for the original test pressure of the plate exchanger process stream.

Foreign References	
Table Name	**Call Value**
PltExchStat	rfEquipment
Nozzle	lstNozzle (rfEquipment, StreamID)

PSVAccess (lstPSVAccess)

PSV Access: the list of required accesses associated with a piece of equipment based on the equipment types derived from the PSV or PSV Location supplied types.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**AccessReqd**	("Access Required", dtString, Required Field (PK), Maximum Length 15)
		From a lookup list, select the type of access required for a PSV or PSV Location.

Foreign References	
Table Name	**Call Value**

PSVStat	rfEquipment

RE_Bearings (lstRE_Bearings)

RE Bearings: the list of bearings associated with a piece of equipment based on the equipment types derived from the Centrifugal Pump or RE Motor supplied types.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**BearingID**	("Bearing ID", dtString, Required Field (PK), Maximum Length 20)
		A unique identifier for each individual or group of bearings.
string	BearingType	("Bearing Type", dtString, Maximum Length 20)
		From a lookup list, select the type or arrangement of bearing(s).
string	Location	("Location", dtString, Maximum Length 35)
		From a lookup list, select the location of the bearing(s).
string	Lubrication	("Lubrication", dtString, Maximum Length 20)
		From a lookup list, select the type of lubrication for the bearings of the pump.
decimal?	OilCapacity	("Oil Capacity", dtDouble, Unit-Based Field)
		The constant level oil preference or proper oil level for the lubrication of the bearings.
string	OilProduct	("Oil Product", dtString, Maximum Length 20)
		From a lookup list, select the oil viscosity or product ISO grade for the lubrication of the bearings.
string	Purpose	("Purpose", dtString, Maximum Length 20)
		From a lookup list, select the purpose of the bearing(s).
int?	Quantity	("Quantity", dtInteger)
		The quantity of similar or specific bearings.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

SectionDesign (lstSectionDesign)

Section Design: the list of sections associated on the Static Data for a piece of equipment.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**SectionID**	("Section ID", dtString, Required Field (PK), Maximum Length 35)
		Enter a unique identifier for a section. Examples: tubes, tube-sheets, floating head, tie rods, baffles, etc or simply a number.
decimal?	AllowStress	("Allowable Stress", dtDouble, Unit-Based Field)

<table>
<tr><td></td><td></td><td>The material stress value for a section, typically a shell course. This can be obtained from the API Data Report form, the construction drawings or the engineering specs.</td></tr>
<tr><td rowspan="2">TG17?</td><td rowspan="2">CalcType</td><td>("Minimum Type", dtInteger)</td></tr>
<tr><td>Indicates how the minimum thickness was entered or calculated. Choices are:
<table>
<tr><th></th><th>Definition</th></tr>
<tr><td>NCA</td><td>Indicates that the Nominal - Corrosion Allowance was used to calculate the Minimum Thickness.</td></tr>
<tr><td>Other</td><td>Indicates that an externally calculated, typically an engineering standard, was used to calculate the Minimum Thickness.</td></tr>
<tr><td>t-Min Code</td><td>Indicates that the Minimum Thickness was user entered.</td></tr>
<tr><td>t-Min Visions</td><td>Indicates that the Visions t-Min calculator was used to calculate Minimum Thickness.
Limitation: This option is not updatable and only available in the Visions Enterprise Client application when the tMin calculator is used.</td></tr>
</table></td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">CJointEff</td><td>("Circumferential Joint Eff", dtDouble, Unit-Based Field)</td></tr>
<tr><td>The circumferential joint efficiency of a section. This can be obtained from the ASME Data Report or the Manufacturer drawings.</td></tr>
<tr><td rowspan="2">long?</td><td rowspan="2">CJointNo</td><td>("Circumferential Joint No.", dtDouble)</td></tr>
<tr><td>The number of circumferential weld joints in a section. This can be obtained from the ASME Data Report or the Manufacturer drawings.</td></tr>
<tr><td rowspan="2">string</td><td rowspan="2">CJointType</td><td>("Circumferential Joint Type", dtString, Maximum Length 20)</td></tr>
<tr><td>From a lookup list, select the type of circumferential weld joint. Example: Fusion, etc. This can be obtained from the ASME Data Report.</td></tr>
<tr><td rowspan="2">string</td><td rowspan="2">Comments</td><td>("Comments", dtString, Maximum Length 255)</td></tr>
<tr><td>Enter any additional remarks or important information relating to section or design of section.</td></tr>
<tr><td rowspan="2">string</td><td rowspan="2">ConstructionMethod</td><td>("Construction Method", dtString, aximum Length 20)</td></tr>
<tr><td>From a lookup list, select how the construction method for the section. This usually indicates how the tubes are attached to the tube-sheets. This may be obtained from the ASME Data Report or the manufacturer construction drawings.</td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">CorrAllow</td><td>("Corr Allow", dtDouble, Unit-Based Field)</td></tr>
<tr><td>The given corrosion allowance of a section. This can be obtained from the ASME Data Report.</td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">Diameter</td><td>("Diameter", dtDouble, Unit-Based Field)</td></tr>
<tr><td>The inside diameter of a section. This can be obtained from the ASME Data Report or manufacturer construction drawings.</td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">Diameter2</td><td>("Diameter 2", dtDouble, Unit-Based Field)</td></tr>
<tr><td>Enter a second diameter value for a section with a non-circular shape. Example: Conical</td></tr>
<tr><td>decimal?</td><td>EngStandThick</td><td>("Eng Standard Thick", dtDouble, Unit-Based Field)</td></tr>
</table>

		The engineering standard thickness for the section.
string	FireproofMaterial	("Fireproofing Material", dtString, Maximum Length 30)
		From a lookup list, select the type of fireproofing material for a section.
decimal?	FireproofThick	("Fireproofing Thick", dtDouble, Unit-Based Field)
		The thickness of the applied fireproofing material for a section.
string	FlangeRating	("Flange Rating", dtString, Maximum Length 10)
		From a lookup list, select the type of ANSI flange rating. This may be obtained from the ASME Data Report, the manufacturer drawings or engineering specifications.
string	HowAttached	("How Attached", dtString, Maximum Length 20)
		From a lookup list, select how the section is attached. This usually indicates how the tubes are attached to the tube-sheets. This can be obtained from the ASME Data Report or the manufacturer construction drawings.
bool?	ImpactTest	("Impact Test Exempt", dtInteger)
		Identify whether the material and welding procedures were exempt from impact testing. Choices are: 0, N, No, 1, Y, Yes. This can be obtained from the ASME Data Report.
string	ImpTestReason	("Exemption Reason", dtString, Maximum Length 25)
		From a lookup list, select the applicable Code paragraph which was utilized to exempt the material and welding procedures from impact testing. This can be obtained from the ASME Data Report.
decimal?	InsideRadius	("Inside Radius", dtDouble, Unit-Based Field)
		The inside radius of a section. It can be important when calculating tubesheet thicknesses in steam drums and muddrums. This can be obtained from the ASME Data Report form or from the Manufacturer drawings.
string	InsulationCode	("Insulation Code", dtString, Maximum Length 20)
		From a lookup list, select the insulation code. Examples: H for Hot, C for Cold, PP for Personnel Protection, NB for Noise Mitigation.
decimal?	InsulationThick	("Insulation Thick", dtDouble, Unit-Based Field)
		The thickness of the installed insulation for a section. This can be obtained from the eqiupment design specifications.
string	InsulationType	("Insulation Type", dtString, Maximum Length 20)
		From a lookup list, select the type of insulation for a section. Example: Fiberglass, Asbestos, etc. This can be obtained from the equipment design specifications.
int?	ItemCounter	("Coating Counter", dtInteger, FK (lstCoatLin))
		Select the cross-reference for the coating, lining or cladding identifier that indicates the protective coating for a section. Note: The list is sourced from the Coatings (lstCoatLin) and must exist before linking to a section.
decimal?	JointEfficiency	("Joint Efficiency", dtDouble, Unit-Based Field)

		The welded joint efficiencies of tube to tube joints (if any), integral channels, etc. This can be obtained from the ASME Data Report or the manufacturer construction drawings.
decimal?	LJointEff	("Longitudinal Joint Eff", dtDouble, Unit-Based Field) The joint efficiency of a section. This can be obtained from the ASME Data Report or the Manufacturer drawings.
long?	LJointNo	("Longitudinal Joint No.", dtDouble) The number of longitudinal weld joints in a section. This can be obtained from the ASME Data Report or the Manufacturer drawings.
string	LJointType	("Longitudinal Joint Type", dtString, Maximum Length 20) From a lookup list, select the type of weld joint of a section. Example: Fusion, etc. This can be obtained from the ASME Data Report.
string	Location	("Location", dtString, Maximum Length 35) From a lookup list, select the location of a section.
string	MaterialID	("Material ID", dtString, Maximum Length 15) A code determined by the User. Usually indicates the basic material and the year of the Code of reference. For example, A105/98 would be for SA-105 material from the 1998 ASME Codes. Note: The values are sourced from the Material Code Lookup Data and the value must exist before updating a line number.
decimal?	MAWT	("MAWT", dtDouble, Unit-Based Field) The Maximum Allowable Working Temperature for a section. This can be obtained from the ASME Data Report or the equipment nameplate.
decimal?	MinThick	("Min Thick", dtDouble, Unit-Based Field) Enter or calculate the ASME Code calculated minimum thickness for the section.
decimal?	NomThick	("Nominal", dtDouble, Unit-Based Field) The actual or nominal wall thickness of a section. This can be obtained from the ASME Data Report.
int?	NumOrQty	("Num or Qty", dtInteger) The number or quantity of a section, i.e.: 4 east waterwall inlet headers. This can be obtained from the ASME Data Report or the Manufacturer drawings.
decimal?	OD	("OD", dtDouble, Unit-Based Field) The outer diameter of a section.
decimal?	OperPress	("Operating Press", dtDouble, Unit-Based Field) The normal operating pressure of a section, in the applicable units. This can be obtained from Operations.
decimal?	OperTemp	("Operating Temp", dtDouble, Unit-Based Field) The normal operating temperature of a section, in the applicable units. This can be obtained from Operations.
bool?	PWHT	("PWHT", dtInteger)

		Identify whether a section was stress relieved during fabrication. Choices are: 0, N, No, 1, Y, Yes. This can be obtained from the ASME Data Report or the equipment stamping.
string	Radiography	("Radiography", dtString, Maximum Length 15) From a lookup list, select the amount of RT completed on a section during construction. This can be obtained from the ASME Data Report or the equipment stamping.
string	Reason	("PWHT Reason", dtString, Maximum Length 25) From a lookup list, select the reason for stress relieving. Normally the choices are Code or Service.
string	SectionPosition	("Section Position", dtString, Maximum Length 15) From a lookup list, select the position of a section.
string	SectionShortName	("Section DL ID", dtString, Fixed Length 6) From a lookup list, select the shortname for a section used in the Visions Datalogger interface when the equipment information is passed to the datalogging unit. Note: The value must use all 4 characters, it is recommended that you pick a prefix as a filler when the value is shorter than 4 characters.
string	SectionType	("Section Type", dtString, Maximum Length 15) From a lookup list, select the general type of a section. Examples: Head, Bottom, Waterwall, Roof, Primary Seal, Floor, Top, etc.
decimal?	SectLength	("Section Length/Height", dtDouble, Unit-Based Field) The overall length or height of a section. This can be obtained from the ASME Data Report or the Manufacturer drawings.
decimal?	THoleLigEffCirc	("Tubehole Lig Eff Circumf", dtDouble) The design circumferential tube hole ligament efficiency. This is used in tubesheet thickness calculations and can be obtained from the ASME Data Report.
decimal?	THoleLigEffLong	("Tubehole Lig Eff Longitudinal", dtDouble) The design longitudinal tube hole ligament efficiency. This is used in tubesheet thickness calculations and can be obtained from the ASME Data Report.
bool?	TubeNonPress	("Tube Non-Press", dtInteger) Indicate whether the tube side of the section is considered non-pressure. Choices are: 0, N, No, 1, Y, Yes. This can be obtained from the Data Report or design specifications.
string	TubeNPReason	("Tube NP Reason", dtString, Maximum Length 25) From a lookup list, select the reason why (vented to atmosphere, etc.) the tubeside is considered non-pressure. This may be obtained from the data report or design specifications.
decimal?	TubePitch	("Tube Pitch", dtDouble, Unit-Based Field) The pitch (distances) between the tubes in the section. This can be obtained from the ASME Data Report or the Manufacturer drawings.
string	TubeType	("Tube Type", dtString, Maximum Length 20)

		From a lookup list, select the type of tubes. Examples: Bare, Membrane, Finned, etc. This may be obtained from the ASME Data Report or the Manufacturer drawings.
string	TypeSect	("Shape Type", dtString, Maximum Length 25)
		From a lookup list, select the description of the shape of the component. For shells this could be: Cylindrical, Cone, etc... This may be obtained from the vessel data report or design specifications.
decimal?	Width	("Width", dtDouble, Unit-Based Field)
		The width of a section.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment
CoatLin	lstCoatLin (rfEquipment, ItemCounter)
GL_CoatLin_V	rfGL_CoatLin_V (rfEquipment, ItemCounter)

TankVolumes (lstTankVolumes)

Tank Volumes: the list of volumes associated with a piece of equipment based on the equipment types derived from the Tank supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
decimal	**TankLevel**	("Tank Level", dtDouble, Required Field (PK), Unit-Based Field)
		The level measurement of a tank for a specific volume. This is used with the tank volume trending to determine projected life of a remaining volume. (Similar to tank strapping measurements)
decimal?	Volume	("Volume", dtDouble, Unit-Based Field)
		The volume of product of a tank for the specified level measurement.

Foreign References	
Table Name	**Call Value**
TankStatic	rfEquipment

TrayPack (lstTrayPack)

Internals: the list of internal trays or packing associated with a piece of equipment based on the equipment types derived from the Exchanger and Vessel supplied types.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**TrayID**	("Tray ID", dtString, Required Field (PK), Maximum Length 20)
		Enter a unique identifier for each tray, pad or packing for a piece of equipment.
string	Description	("Description", dtString, Maximum Length 50)
		Enter a description for each tray, pad or packing for a piece of equipment.

string	Manufacturer	("Manufacturer", dtString, Maximum Length 50)
		From a lookup list, select the name of the manufacturer of the trays, pads or packing. This can be obtained from the construction drawings or the installation documents or specifications.
string	Material	("Material", dtString, Maximum Length 15)
		From a lookup list, select the type of material for the tray, pad or packing. This can be obtained from the construction drawings.
string	Packing	("Packing", dtString, Maximum Length 15)
		From a lookup list, select the type of packing installed. Example: ratchet rings, ceramic pieces, catalyst, etc. This can be obtained from the construction drawings.
int?	Quantity	("Quantity", dtInteger)
		The number or quantity of trays or pads. This can be obtained from the construction drawings.
decimal?	Thickness	("Thick", dtDouble, Unit-Based Field)
		The thickness of the trays, pads or packing. This can be obtained from the construction drawings.
string	TrayType	("Tray Type", dtString, Maximum Length 30)
		From a lookup list, select the type of tray or pad. Example: chimney, bubble cap, sieve, etc. This can be obtained from the construction drawings.

Foreign References	
Table Name	**Call Value**
StatHdr	rfEquipment

TubeData (lstTubeData)

Tube Data: the list of tube sets associated with a piece of equipment based on the equipment types derived from the Firetube Boiler supplied type.

Data Type	**Field**	**Field Information (GUI Reference, GUI Data Type, Size, Purpose)**
int	**SetNumber**	("Set Number", dtInteger, Required Field (PK))
		Enter a unique set number for each tube configuration in a firetube boiler.
decimal?	HeatSurface	("Heat Surface", dtDouble, Unit-Based Field)
		The heating surface for tubes in a firetube boiler.
string	HowAttachedColdEnd	("How Attached - Cold End", dtString, Maximum Length 20)
		From a lookup list, select how the cold end of the tubes are attached in a firetube boiler.
string	HowAttachedHotEnd	("How Attached - Hot End", dtString, Maximum Length 20)
		From a lookup list, select how the hot end of the tubes are attached in a firetube boiler.
		("Material", dtString, Maximum Length 15)

string	Material	From a lookup list, select the type of material for the tubes in a firetube boiler.
decimal?	OuterDiameter	("OD", dtDouble, Unit-Based Field)
		The outer diameter for the tubes in a firetube boiler.
decimal?	Pitch	("Pitch", dtDouble, Unit-Based Field)
		The pitch of the tubes in a firetube boiler.
int?	Quantity	("Quantity", dtInteger)
		The quantity of tubes with a similar configuration in a firetube boiler.
string	SectionType	("Section Type", dtString, Maximum Length 15)
		From a lookup list, select the type of section for the tubes in a firetube boiler.
decimal?	TubeLength	("Tube Length", dtDouble, Unit-Based Field)
		The length for the tubes in a firetube boiler.
string	TubeType	("Tube Type", dtString, Maximum Length 20)
		From a lookup list, select the type of tubes in a firetube boiler.
decimal?	WallThickness	("Wall Thick", dtDouble, Unit-Based Field)
		The thickness of the wall for the tubes in a firetube boiler.

Foreign References	
Table Name	**Call Value**
FTBoilStat	rfEquipment

ValveAccess (lstValveAccess)

Valve Accessories: the list of accessories associated with a piece of equipment based on the equipment types derived from the Valve supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**VlvAccessId**	("Accessory ID", dtString, Required Field (PK), Maximum Length 15)
		A system id that uniquely identifies each accessory for a valve.
string	Description	("Description", dtString, Maximum Length 255)
		Enter a description for each accessory of a valve.
string	VlvAccessType	("Accessory Type", dtString, Maximum Length 15)
		From a lookup list, select the type of accessory for a valve.

Foreign References	
Table Name	**Call Value**
ValveStat	rfEquipment

ValveLimitSwtch (lstValveLimitSwtch)

Valve Limit Switches: the list of limit switches associated with a piece of equipment based on the equipment types derived from the Valve supplied type.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**LimitSwitchId**	("Limit Switch ID", dtString, Required Field (PK), Maximum Length 15)
		A system id that uniquely identifies each limit switch for a valve.
decimal?	ContactRate	("Contact Rating", dtDouble, Unit-Based Field)
		The contact rating for a limit switch of a valve.
int?	LSQuantity	("Quantity", dtInteger)
		The quantity of similar limit switches for a valve.
string	Manufacturer	("Manufacturer", dtString, Maximum Length 50)
		The manufacturer for a limit switch of a valve.
string	ModelNo	("Model Number", dtString, Maximum Length 50)
		The model number for a limit switch of a valve.
string	SwitchPosition	("Switch Position", dtString, Maximum Length 15)
		From a lookup list, select the position of a limit switch of a valve.
string	SwitchType	("Switch Type", dtString, Maximum Length 15)
		From a lookup list, select the type of switch for a limit switch for a valve.
string	TagNumber	("Tag Number", dtString, Maximum Length 15)
		The tag number for a limit switch for a valve.

Foreign References	
Table Name	**Call Value**
ValveStat	rfEquipment

Call Examples

CreateEquipment

Using the Create Equipment call, the following examples provide a guideline using the identified call conventions.

Call with Debug

```
<!-- This example demonstrates the CreateEquipment call providing the key values as SOAP
    parameters. It can create one piece of equipment. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:CreateEquipment>
```

```
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>D80-HC-21</met:EquipNum>
    <met:PlantNum>U-3000 – Demonstration DB</met:PlantNum>
    <met:EquipType>Boiler</met:EquipType>
    <met:Values>
    [Values]
       [DebugInfo /]
       [Value field="OutageExtent"]Plant[/Value]
       [Value field="ERPNumber"]00004818[/Value]
       [Value field="rfDesMat.OperatingLimit"]76[/Value]
       [Value field="rfBoilerStatic.Other5"]Approved by CSRI[/Value]
       [Value field="rfBoilerStatic.ShopFabricated"]false[/Value]
    [/Values]
    </met:Values>
   </met:CreateEquipment>
 </soapenv:Body>
</soapenv:Envelope>
```

Call with Keys as an Array

```
<!-- This example shows CreateEquipment with all the key values rolled into the values
    array, allowing a single call to create more than one piece of equipment. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:CreateEquipment>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:Values>
    [Records]
    [Values]
       [DebugInfo /]
       [Value field="EquipNum"]D80-HC-21[/Value]
       [Value field="PlantNum"]U-3000 – Demonstration DB[/Value]
       [Value field="EquipType"]Boiler[/Value]
       [Value field="ERPNumber"]00004818[/Value]
       [Value field="FlowDiagram"]34324-F[/Value]
    [/Values]
    [Values]
       [Value field="EquipNum"]D80-HC-22[/Value]
       [Value field="PlantNum"]U-3000 – Demonstration DB[/Value]
       [Value field="EquipType"]Boiler[/Value]
       [Value field="ERPNumber"]00004819[/Value]
       [Value field="FlowDiagram"]34324-F[/Value]
    [/Values]
    [/Records]
    </met:Values>
   </met:CreateEquipment>
 </soapenv:Body>
</soapenv:Envelope>
```

Response with Debug Information

```
<!-- A response from VisAPI looks like this. The data is echoed back in the response
    message to allow troubleshooting due to the ‘DebugInfo’ in the above example. -->
```

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <CreateEquipmentResponse xmlns="http://metegrity.com">
    <CreateEquipmentResult>true</CreateEquipmentResult>
    <ErrorMsg>Operation successful.
   ----
   [?xml version="1.0" encoding="utf-16"?]
   [DataArray count="1" p1:xsi="&amp;quot;http://www.w3.org/2001/XMLSchema-
   instance&amp;quot;" xmlns:p1="xmlns"]
   [DataClass p3:type="Metegrity.Data.StatHdr" xsi:type="StatHdr" ID="1000"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:p3="xsi"]
   [TF]*[/TF]
   [EquipmentID gen="EQUIPMENTID" key="EQUIPMENTID" Order="5" /]
   [EquipmentID]V5DI0000000IJSS[/EquipmentID]
   [UnitID]Metegrity Refinery[/UnitID]
   [PlantNum]U-3000 - Demonstration DB[/PlantNum]
   [EquipType]Boiler[/EquipType]
   [EquipNum]D80-HC-21[/EquipNum]
   [ERPNumber]00004818[/ERPNumber]
   [OutageExtent]Plant[/OutageExtent]
   [InService]true[/InService]
   [SubEquipment]302[/SubEquipment]
   [DeleteFlag]false[/DeleteFlag]
   [rfBoilerStatic]
   [DataClass xsi:type="BoilerStatic" ID="1001"]
   [TF]*[/TF]
   [EquipmentID byproxy="true" Order="5"]
   [EQUIPMENTID table="backref"]EQUIPMENTID[/EQUIPMENTID]
   [/EquipmentID]
   [EquipmentID]V5DI0000000IJSS[/EquipmentID]
   [ShopFabricated]false[/ShopFabricated]
   [Other5]Approved by CSRI[/Other5]
   [/DataClass]
   [/rfBoilerStatic]
   [rfDesMat]
   [DataClass xsi:type="DesMat" ID="1002"]
   [TF]*[/TF]
   [EquipmentID byproxy="true" Order="5"]
   [EQUIPMENTID table="backref"]EQUIPMENTID[/EQUIPMENTID]
   [/EquipmentID]
   [EquipmentID]V5DI0000000IJSS[/EquipmentID]
   [OperatingLimit]76[/OperatingLimit]
   [/DataClass]
   [/rfDesMat]
   [rfSchedule]
   [DataClass xsi:type="Schedule" ID="1003"]
   [TF]*[/TF]
   [EquipmentID byproxy="true" Order="5"]
   [EQUIPMENTID table="backref"]EQUIPMENTID[/EQUIPMENTID]
   [/EquipmentID]
   [EquipmentID]V5DI0000000IJSS[/EquipmentID]
   [ScheduleFlag]false[/ScheduleFlag]
   [LastInServFlag]1[/LastInServFlag]
   [/DataClass]
   [/rfSchedule]
   [/DataClass]
   [/DataArray]
   ----</ErrorMsg>
   </CreateEquipmentResponse>
 </s:Body>
</s:Envelope>
```

Response without Debug Information

```
<!-- Without DebugInfo, the response looks like this. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <CreateEquipmentResponse xmlns="http://metegrity.com">
    <CreateEquipmentResult>true</CreateEquipmentResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </CreateEquipmentResponse>
 </s:Body>
</s:Envelope>
```

UpdateEquipment

Call without Debug

```
<!-- This example sets the Flow Diagram fields for D80-HC-21 and D80-HC-22
 to NF-7524 and NF-7525 accordingly. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:UpdateEquipment>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:Values>
    [Records]
    [Values]
      [DebugInfo /]
      [Value field="EquipNum"]D80-HC-21[/Value]
      [Value field="PlantNum"]U-3000 - Demonstration DB[/Value]
      [Value field="EquipType"]Boiler[/Value]
      [Value field="FlowDiagram"]NF-7524[/Value]
    [/Values]
    [Values]
      [Value field="EquipNum"]D80-HC-22[/Value]
      [Value field="PlantNum"]U-3000 - Demonstration DB[/Value]
      [Value field="EquipType"]Boiler[/Value]
      [Value field="FlowDiagram"]NF-7525[/Value]
    [/Values]
    [/Records]
    </met:Values>
   </met:UpdateEquipment>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- A response from UpdateEquipment. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <UpdateEquipmentResponse xmlns="http://metegrity.com">
    <UpdateEquipmentResult>true</UpdateEquipmentResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </UpdateEquipmentResponse>
 </s:Body>
</s:Envelope>
```

UpdateSchedule

Call without Debug

```
<!-- Set the Due Date for D80-HC-22 to 07-08-2016. This is just a logical shortcut for
    calling UpdateEquipment, which can change rfSchedule fields. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:UpdateSchedule>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>D80-HC-22</met:EquipNum>
    <met:PlantNum>U-3000 - Demonstration DB</met:PlantNum>
    <met:EquipType>Boiler</met:EquipType>
    <met:DueDate>07-08-2016</met:DueDate>
    <met:DateFmt>MM-dd-yyyy</met:DateFmt>
   </met:UpdateSchedule>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- Schedule change successful. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <UpdateScheduleResponse xmlns="http://metegrity.com">
    <UpdateScheduleResult>true</UpdateScheduleResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </UpdateScheduleResponse>
 </s:Body>
</s:Envelope>
```

DeleteEquipment

Call without Debug

```
<!-- This example demonstrates the use of the DeleteEquipment API call, placing the
    indicated equipment in the Visions Recycle Bin. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:DeleteEquipment>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>D80-HC-21</met:EquipNum>
    <met:PlantNum>U-3000 - Demonstration DB</met:PlantNum>
    <met:EquipType>Boiler</met:EquipType>
   </met:DeleteEquipment>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- This is a typical response message from DeleteEquipment. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <DeleteEquipmentResponse xmlns="http://metegrity.com">
    <DeleteEquipmentResult>true</DeleteEquipmentResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </DeleteEquipmentResponse>
 </s:Body>
</s:Envelope>
```

RemoveEquipment

Call without Debug

```
<!-- This is a typical call to RemoveEquipment, which purges an asset from the database
    entirely. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:RemoveEquipment>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>D80-HC-21</met:EquipNum>
    <met:PlantNum>U-3000 - Demonstration DB</met:PlantNum>
    <met:EquipType>Boiler</met:EquipType>
   </met:RemoveEquipment>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- Because we used the same equipment earlier in DeleteEquipment, the response
    indicates that the VisAPI can't find it. Items in the Recycle Bin are invisible to
    VisAPI. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <RemoveEquipmentResponse xmlns="http://metegrity.com">
    <RemoveEquipmentResult>false</RemoveEquipmentResult>
    <ErrorMsg>Specified equipment record 'D80-HC-21/U-3000 - Demonstration DB/Boiler' not
   found.</ErrorMsg>
   </RemoveEquipmentResponse>
 </s:Body>
</s:Envelope>
```

MoveEquipment

Call without Debug

```
<!-- Here, we use MoveEquipment to move D80-HC-21 to the Dehydration plant. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:MoveEquipment>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>D80-HC-21</met:EquipNum>
    <met:PlantNum>U-3000 - Demonstration DB</met:PlantNum>
    <met:EquipType>Boiler</met:EquipType>
    <met:NewPlantNum>Plant 10 - Dehydration</met:NewPlantNum>
   </met:MoveEquipment>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The MoveEquipment response shows us that the transfer was successful. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <MoveEquipmentResponse xmlns="http://metegrity.com">
    <MoveEquipmentResult>true</MoveEquipmentResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </MoveEquipmentResponse>
 </s:Body>
</s:Envelope>
```

RenumberEquipment

Call without Debug

```
<!-- This message demonstrates the renaming of D80-HC-22 to D80-HC22A using the
    RenumberEquipment API. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:RenumberEquipment>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>D80-HC-22</met:EquipNum>
    <met:PlantNum>U-3000 - Demonstration DB</met:PlantNum>
    <met:EquipType>Boiler</met:EquipType>
    <met:NewEquipNum>D80-HC-22A</met:NewEquipNum>
   </met:RenumberEquipment>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The response indicates the operation succeeded. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <RenumberEquipmentResponse xmlns="http://metegrity.com">
    <RenumberEquipmentResult>true</RenumberEquipmentResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </RenumberEquipmentResponse>
 </s:Body>
</s:Envelope>
```

EquipmentExistsQuery

Call without Debug

```
<!-- This call uses EquipmentExistsQuery to determine if a specific piece of equipment
    exists -- in this case, the Boiler numbered D80-HC-22 in plant U-3000 - Demonstration
    DB. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:EquipmentExistsQuery>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>D80-HC-22</met:EquipNum>
    <met:PlantNum>U-3000 - Demonstration DB</met:PlantNum>
    <met:EquipType>Boiler</met:EquipType>
   </met:EquipmentExistsQuery>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The response says "Operation successful." because the query succeeded, even though
    the equipment it question does not exist (so the call returns false). -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <EquipmentExistsQueryResponse xmlns="http://metegrity.com">
    <EquipmentExistsQueryResult>true</EquipmentExistsQueryResult>
    <existsResult>false</existsResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </EquipmentExistsQueryResponse>
 </s:Body>
</s:Envelope>
```

QueryEquipment

Call without Debug

```
<!-- This call is used to return a specific field value from a given piece of equipment -
    - in this case, the Paint Code for D80-HC-22A. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:QueryEquipment>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>D80-HC-22A</met:EquipNum>
    <met:PlantNum>U-3000 - Demonstration DB</met:PlantNum>
    <met:EquipType>Boiler</met:EquipType>
    <met:FieldPath>rfDesMat.PaintCode</met:FieldPath>
   </met:QueryEquipment>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The PaintCode for D80-HC-22A is apparently PC-1. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <QueryEquipmentResponse xmlns="http://metegrity.com">
    <QueryEquipmentResult>true</QueryEquipmentResult>
    <FieldVal>PC-1</FieldVal>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </QueryEquipmentResponse>
 </s:Body>
</s:Envelope>
```

EquipmentListQuery

Call without Debug

```
<!-- This query selects the Equipment Index fields for all the equipment in Plants U-
    1100, U-1300, U-2000 and U-3000 where the InService flag is set to true. It returns
    the first page with a pageSize of 20, so up to 20 individual records. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com"
   xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
 <soapenv:Header/>
 <soapenv:Body>
   <met:EquipmentListQuery>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>D80-HC-22</met:EquipNum>
    <met:PlantNum>U-3000 - Demonstration DB</met:PlantNum>
    <met:EquipType>Boiler</met:EquipType>
    <met:FieldName1>PlantNum</met:FieldName1>
    <met:FieldValues1>
       <arr:string>U-1100 - Feed Dryers</arr:string>
       <arr:string>U-1300 - Polymerization</arr:string>
       <arr:string>U-2000 - General Information</arr:string>
       <arr:string>U-3000 - Demonstration DB</arr:string>
    </met:FieldValues1>
    <met:FieldName2>InService</met:FieldName2>
    <met:FieldValues2>
       <arr:string>1</arr:string>
    </met:FieldValues2>
    <met:DateFmt>MM-dd-yyyy</met:DateFmt>
    <!--Optional:-->
    <met:ColumnSet>
        rfSuperSubLink_EquipmentID=>SSL;rfMajorEquipment=>ME;rfSchedule=>S;
        rfDesMat=>D;
        UnitID,PlantNum,TrainNum,EquipType,EquipNum,EquipName,Location,ERPNumber,
        RegistrationNo,InService,Status,OutageExtent,ManufSerialNo,D.MAWP,D.TempMAWT,
        D.MDMT,D.MDMTPress,D.OperatingPressure,D.OperatingTemp,S.InspRiskCode,
        S.RestrictedInterval,S.NextInspFullDue,S.NextExtDueDate,S.RLFactor,
        S.OutageDate,D.PaintCode,SSL.ME.EquipNum:Associated_Major,ApplicationType,
        S.BasedOn,S.PointExpireDate,S.LastExtDate,S.ExtInterval,S.InServiceDate,
        S.InstallDate,S.CAApplied,S.CARiskCode
    </met:ColumnSet>
    <met:pageSize>20</met:pageSize>
    <met:pageNum>0</met:pageNum>
   </met:EquipmentListQuery>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- A (comparatively short) example of an EquipmentListQuery response.The zero-index
    entry in the OutList array has the system field labels for returned fields, and the
    moreLeft flag indicates whether the requested page is the last of the records. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <EquipmentListQueryResponse xmlns="http://metegrity.com">
    <EquipmentListQueryResult>true</EquipmentListQueryResult>
    <OutList xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
       xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
       <a:ArrayOfstring>
        <a:string>Unit ID</a:string>
        <a:string>Plant ID</a:string>
        <a:string>Train / Unit</a:string>
        <a:string>Equipment Type</a:string>
        <a:string>Equipment Number</a:string>
        <a:string>Equipment Name</a:string>
        <a:string>Location</a:string>
        <a:string>ERP Number</a:string>
        <a:string>Registration Number</a:string>
        <a:string>In Service</a:string>
        <a:string>Status</a:string>
        <a:string>Outage Extent</a:string>
        <a:string>Mfr Serial No</a:string>
        <a:string>MAWP</a:string>
        <a:string>MAWT</a:string>
        <a:string>MDMT</a:string>
        <a:string>MDMT Press</a:string>
        <a:string>Operating Press</a:string>
        <a:string>Operating Temp</a:string>
        <a:string>IRC</a:string>
        <a:string>Restricted Interval</a:string>
        <a:string>Next Full Insp Due</a:string>
        <a:string>Next Ext Due Date</a:string>
        <a:string>RL Factor</a:string>
        <a:string>Outage Date</a:string>
        <a:string>Paint Code</a:string>
        <a:string>Associated Major</a:string>
        <a:string>Application Type</a:string>
        <a:string>Based On</a:string>
        <a:string>Equipment RLF Date</a:string>
        <a:string>Last External Date</a:string>
        <a:string>External Interval</a:string>
        <a:string>In Service Date</a:string>
        <a:string>Install Date</a:string>
        <a:string>CA Applied</a:string>
        <a:string>Criticality IRC</a:string>
       </a:ArrayOfstring>
       <a:ArrayOfstring>
        <a:string>Metegrity Refinery</a:string>
        <a:string>U-3000 - Demonstration DB</a:string>
        <a:string>Unit 1000</a:string>
        <a:string>Channel</a:string>
        <a:string>D08-E-100B</a:string>
        <a:string>Channel for Feed/Effluent Exchanger 08-E-100B</a:string>
        <a:string>Grade, East of Piperack</a:string>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string>True</a:string>
        <a:string>In Service</a:string>
        <a:string>Equipment</a:string>
        <a:string>EXCH-876CHAN</a:string>
        <a:string>635</a:string>
        <a:string>374</a:string>
        <a:string>-20</a:string>
        <a:string>635.889</a:string>
        <a:string>300</a:string>
        <a:string>300</a:string>
        <a:string>4</a:string>
        <a:string>72</a:string>
        <a:string>01/01/2006</a:string>
        <a:string i:nil="true"/>
        <a:string>0.5</a:string>
        <a:string i:nil="true"/>
        <a:string>PC-1</a:string>
        <a:string>D08-E-100B</a:string>
        <a:string>Feed/Effluent Exchanger</a:string>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
       </a:ArrayOfstring>
    </OutList>
    <moreLeft>true</moreLeft>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </EquipmentListQueryResponse>
 </s:Body>
</s:Envelope>
```

Call without Plant or Equipment Type Filtering

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com"
   xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
<soapenv:Header/>
<soapenv:Body>
   <met:EquipmentListQuery>
   <met:SiteName>[Visions SiteName]</met:SiteName>
   <met:UserName>[Visions UserName]</met:UserName>
   <met:Password>[Visions Password]</met:Password>
   <met:ColumnSet>
       rfSuperSubLink_EquipmentID=>SSL;rfMajorEquipment=>ME;rfSchedule=>S;rfDesMat=>D;
       UnitID,PlantNum,TrainNum,EquipType,EquipNum,EquipName,Location,ERPNumber,
       RegistrationNo,InService,Status,OutageExtent,ManufSerialNo,D.MAWP,D.TempMAWT,
       D.MDMT,D.MDMTPress,D.OperatingPressure,D.OperatingTemp,S.InspRiskCode,
       S.RestrictedInterval,S.NextInspFullDue,S.NextExtDueDate,S.RLFactor,S.OutageDate,
       D.PaintCode,SSL.ME.EquipNum:Associated_Major,ApplicationType,S.BasedOn,
       S.PointExpireDate,S.LastExtDate,S.ExtInterval,S.InServiceDate,S.InstallDate,
       S.CAApplied,S.CARiskCode
   </met:ColumnSet>
   <met:pageSize>20</met:pageSize>
   <met:pageNum>0</met:pageNum>
   </met:EquipmentListQuery>
</soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <EquipmentListQueryResponse xmlns="http://metegrity.com">
    <EquipmentListQueryResult>true</EquipmentListQueryResult>
    <OutList xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
      xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
      <a:ArrayOfstring>
       <a:string>Unit ID</a:string>
       <a:string>Plant ID</a:string>
       <a:string>Train / Unit</a:string>
       <a:string>Equipment Type</a:string>
       <a:string>Equipment Number</a:string>
       <a:string>Equipment Name</a:string>
       <a:string>Location</a:string>
       <a:string>ERP Number</a:string>
       <a:string>Registration Number</a:string>
       <a:string>In Service</a:string>
       <a:string>Status</a:string>
       <a:string>Outage Extent</a:string>
       <a:string>Mfr Serial No</a:string>
       <a:string>MAWP</a:string>
       <a:string>MAWT</a:string>
       <a:string>MDMT</a:string>
       <a:string>MDMT Press</a:string>
       <a:string>Operating Press</a:string>
       <a:string>Operating Temp</a:string>
       <a:string>IRC</a:string>
       <a:string>Restricted Interval</a:string>
       <a:string>Next Full Insp Due</a:string>
       <a:string>Next Ext Due Date</a:string>
       <a:string>RL Factor</a:string>
       <a:string>Outage Date</a:string>
       <a:string>Paint Code</a:string>
       <a:string>Associated Major</a:string>
       <a:string>Application Type</a:string>
       <a:string>Based On</a:string>
       <a:string>Equipment RLF Date</a:string>
       <a:string>Last External Date</a:string>
       <a:string>External Interval</a:string>
       <a:string>In Service Date</a:string>
       <a:string>Install Date</a:string>
       <a:string>CA Applied</a:string>
       <a:string>Criticality IRC</a:string>
      </a:ArrayOfstring>
      <a:ArrayOfstring>
       <a:string>Metegrity Pulp &amp; Paper</a:string>
       <a:string>Plant 31 - Power Generation</a:string>
       <a:string i:nil="true"/>
       <a:string>PSV</a:string>
       <a:string>235989-C-546</a:string>
       <a:string>Superheater Outlet PSV</a:string>
```

```
        <a:string>Boiler House</a:string>
        <a:string>5548321</a:string>
        <a:string>356987</a:string>
        <a:string>True</a:string>
        <a:string>In Service</a:string>
        <a:string>Plant</a:string>
        <a:string>235989-C-546</a:string>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string>13</a:string>
        <a:string>36</a:string>
        <a:string>29/10/2017</a:string>
        <a:string>29/04/2016</a:string>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string>Safety</a:string>
        <a:string>RI</a:string>
        <a:string i:nil="true"/>
        <a:string>29/10/2014</a:string>
        <a:string>18</a:string>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
       </a:ArrayOfstring>
    </OutList>
    <moreLeft>true</moreLeft>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </EquipmentListQueryResponse>
 </s:Body>
</s:Envelope>
```

CreateLink

Call without Debug

```
<!-- This SOAP call uses the CreateLink API to create a Protection Link between the
    Boiler D80-HC-21 and PSV 08-PSV-101. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:CreateLink>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:Class>Protection</met:Class>
    <met:Values>
    [Values]
    [Value field='TypeOfLink']Safety[/Value]
    [/Values]
    </met:Values>
    <met:KeyA1>U-3000 - Demonstration DB</met:KeyA1>
    <met:KeyA2>Boiler</met:KeyA2>
    <met:KeyA3>D80-HC-21</met:KeyA3>
    <met:KeyB1>Plant 08 - ULSD Hydrotreater</met:KeyB1>
    <met:KeyB2>PSV</met:KeyB2>
    <met:KeyB3>08-PSV-101</met:KeyB3>
   </met:CreateLink>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The typical response indicates the link was created correctly. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <CreateLinkResponse xmlns="http://metegrity.com">
    <CreateLinkResult>true</CreateLinkResult>
    <ErrorMsg>Operation complete.</ErrorMsg>
   </CreateLinkResponse>
 </s:Body>
</s:Envelope>
```

Invalid Call without Debug

Note:An attempt to create a CP protection link with incorrect equipment types.

```
<!-- This SOAP call uses the CreateLink API to create a CP Protection Link between the
Boiler D80-HC-21 and PSV 08-PSV-101 -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:CreateLink>
   <met:SiteName>NewDev5</met:SiteName>
   <met:UserName>usera</met:UserName>
   <met:Password>****</met:Password>
   <met:Class>CPProtection</met:Class>
   <met:KeyA1>U-3000 - Demonstration DB</met:KeyA1>
   <met:KeyA2>Boiler</met:KeyA2>
   <met:KeyA3>D80-HC-21</met:KeyA3>
   <met:KeyB1>Plant 08 - ULSD Hydrotreater</met:KeyB1>
   <met:KeyB2>PSV</met:KeyB2>
   <met:KeyB3>08-PSV-101</met:KeyB3>
   </met:CreateLink>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
 <CreateLinkResponse xmlns="http://metegrity.com">
<CreateLinkResult>false</CreateLinkResult>
<ErrorMsg>The Visions API encountered an error processing your call.
----
Error: Cathodic Protection Link equipment types mismatch.
----</ErrorMsg>
 </CreateLinkResponse>
 </s:Body>
</s:Envelope>
```

UpdateLink

Call without Debug

```
<!-- We can use UpdateLink to change the TypeOfLink value for the example link created
    above. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:UpdateLink>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:Class>Protection</met:Class>
    <met:Values>
    [Values]
    [DebugInfo /]
    [Value field="TypeOfLink"]Injection[/Value]
    [/Values]
    </met:Values>
    <met:KeyA1>U-3000 - Demonstration DB</met:KeyA1>
    <met:KeyA2>Boiler</met:KeyA2>
    <met:KeyA3>D80-HC-21</met:KeyA3>
    <met:KeyB1>Plant 08 - ULSD Hydrotreater</met:KeyB1>
    <met:KeyB2>PSV</met:KeyB2>
    <met:KeyB3>08-PSV-101</met:KeyB3>
   </met:UpdateLink>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The response indicates the update succeeded. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <UpdateLinkResponse xmlns="http://metegrity.com">
    <UpdateLinkResult>true</UpdateLinkResult>
    <ErrorMsg>Operation complete.</ErrorMsg>
   </UpdateLinkResponse>
 </s:Body>
</s:Envelope>
```

CheckForLink

Call without Debug

```
<!-- The CheckForLink API tells us whether a link of a specified type between two Visions
    entities exists. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:CheckForLink>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:Class>Protection</met:Class>
    <met:KeyA1>U-3000 - Demonstration DB</met:KeyA1>
    <met:KeyA2>Boiler</met:KeyA2>
    <met:KeyA3>D80-HC-21</met:KeyA3>
    <met:KeyB1>Plant 08 - ULSD Hydrotreater</met:KeyB1>
    <met:KeyB2>PSV</met:KeyB2>
    <met:KeyB3>08-PSV-101</met:KeyB3>
   </met:CheckForLink>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The response is fairly typical. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <CheckForLinkResponse xmlns="http://metegrity.com">
    <CheckForLinkResult>true</CheckForLinkResult>
    <ErrorMsg>Operation in progress...</ErrorMsg>
    <IsLinkPresent>true</IsLinkPresent>
   </CheckForLinkResponse>
 </s:Body>
</s:Envelope>
```

RemoveLink

Call without Debug

```
<!-- The RemoveLink API deletes an existing link, as shown. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:RemoveLink>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:Class>Protection</met:Class>
    <met:KeyA1>U-3000 - Demonstration DB</met:KeyA1>
    <met:KeyA2>Boiler</met:KeyA2>
    <met:KeyA3>D80-HC-21</met:KeyA3>
    <met:KeyB1>Plant 08 - ULSD Hydrotreater</met:KeyB1>
    <met:KeyB2>PSV</met:KeyB2>
    <met:KeyB3>08-PSV-101</met:KeyB3>
   </met:RemoveLink>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- This response indicates that the link was removed correctly. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <RemoveLinkResponse xmlns="http://metegrity.com">
    <RemoveLinkResult>true</RemoveLinkResult>
   <ErrorMsg>Operation complete.</ErrorMsg>
   </RemoveLinkResponse>
 </s:Body>
</s:Envelope>
```

Change Pipeline Status

It is treated as a virtual field of StatHdr; because it is a virtual field, it has a hash-sign prepended to it at the start.

Call without Debug

```
<!-- The Change deletes an existing link, as shown. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:met="http://metegrity.com">
   <soapenv:Header/>
   <soapenv:Body>
      <met:UpdateEquipment>
         <met:SiteName>[Visions SiteName]</met:SiteName>
         <met:UserName>[Visions UserName]</met:UserName>
         <met:Password>[Visions Password]</met:Password>
         <met:Values>
           [Records]
             [Values]
                [Value field="EquipNum"]PLS2[/Value]
                [Value field="PlantNum"]Crystals Plant[/Value]
                [Value field="EquipType"]Pipeline Segment[/Value]
                [Value field="Status"]Suspended[/Value]
                [Value field="#StatusComments"]Changed Through VisAPI[/Value]
             [/Values]
           [/Records]
         </met:Values>
      </met:UpdateEquipment>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Body>
      <UpdateEquipmentResponse xmlns="http://metegrity.com">
         <UpdateEquipmentResult>true</UpdateEquipmentResult>
         <ErrorMsg>Operation successful.</ErrorMsg>
      </UpdateEquipmentResponse>
   </s:Body>
</s:Envelope>
```

CircuitEquipQuery

Call without Debug

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:met="http://metegrity.com"
xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
   <soapenv:Header/>
   <soapenv:Body>
      <met:CircuitEquipQuery>
         <met:SiteName>[Visions SiteName]</met:SiteName>
         <met:UserName>[Visions UserName]</met:UserName>
         <met:Password>[Visions Password]</met:Password>
         <met:CircuitNum>10-0006-X013</met:CircuitNum>
         <met:PlantNum>1006</met:PlantNum>
         <met:EquipType>Circuit</met:EquipType>
      </met:CircuitEquipQuery>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Body>
      <CircuitEquipQueryResponse xmlns="http://metegrity.com">
         <CircuitEquipQueryResult>true</CircuitEquipQueryResult>
           <OutList    xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <a:ArrayOfstring>
               <a:string>Plant ID</a:string>
               <a:string>Train / Unit</a:string>
               <a:string>Equipment Type</a:string>
               <a:string>Equipment Number</a:string>
               <a:string>Equipment Name</a:string>
               <a:string>Circuit Plant</a:string>
               <a:string>Circuit Type</a:string>
               <a:string>Circuit Number</a:string>
               <a:string>Avail on IR</a:string>
            </a:ArrayOfstring>
            <a:ArrayOfstring>
               <a:string>1006</a:string>
               <a:string i:nil="true"/>
               <a:string>Pipe</a:string>
               <a:string>10-0006-X013-J02</a:string>
               <a:string>18IN UPSTREAM OF IP to DOWNSTREAM IP LIMIT</a:string>
               <a:string>1006</a:string>
               <a:string>Circuit</a:string>
               <a:string>10-0006-X013</a:string>
               <a:string>False</a:string>
            </a:ArrayOfstring>
            <a:ArrayOfstring>
               <a:string>1006</a:string>
               <a:string i:nil="true"/>
               <a:string>Pipe</a:string>
               <a:string>10-0006-X013-J01</a:string>
               <a:string>18IN UPSTREAM OF IP to DOWNSTREAM IP LIMIT</a:string>
               <a:string>1006</a:string>
               <a:string>Circuit</a:string>
               <a:string>10-0006-X013</a:string>
               <a:string>False</a:string>
            </a:ArrayOfstring>
            <a:ArrayOfstring>
               <a:string>1006</a:string>
               <a:string i:nil="true"/>
               <a:string>Pipe</a:string>
               <a:string>10-0006-X013-010</a:string>
               <a:string>PLANT 61 PLOT LIMIT to TMPL PUMP/AIRPORT</a:string>
               <a:string>1006</a:string>
               <a:string>Circuit</a:string>
               <a:string>10-0006-X013</a:string>
               <a:string>False</a:string>
            </a:ArrayOfstring>
            <a:ArrayOfstring>
               <a:string>1006</a:string>
               <a:string i:nil="true"/>
               <a:string>Pipe</a:string>
               <a:string>10-0006-X013-030</a:string>
               <a:string>FSII DIEGME IP to PLANT 10</a:string>
               <a:string>1006</a:string>
               <a:string>Circuit</a:string>
               <a:string>10-0006-X013</a:string>
               <a:string>False</a:string>
            </a:ArrayOfstring>
            <a:ArrayOfstring>
               <a:string>1006</a:string>
               <a:string i:nil="true"/>
               <a:string>Pipe</a:string>
               <a:string>10-0006-X013-J03</a:string>
               <a:string>18IN UPSTREAM OF IP to DOWNSTREAM IP LIMIT</a:string>
               <a:string>1006</a:string>
               <a:string>Circuit</a:string>
               <a:string>10-0006-X013</a:string>
               <a:string>False</a:string>
            </a:ArrayOfstring>
            <a:ArrayOfstring>
               <a:string>1006</a:string>
               <a:string i:nil="true"/>
               <a:string>Pipe</a:string>
               <a:string>10-0006-X013-J04</a:string>
               <a:string>18IN UPSTREAM OF IP to DOWNSTREAM IP LIMIT</a:string>
               <a:string>1006</a:string>
               <a:string>Circuit</a:string>
               <a:string>10-0006-X013</a:string>
               <a:string>False</a:string>
            </a:ArrayOfstring>
         </OutList>
         <moreLeft>false</moreLeft>
         <ErrorMsg>Operation successful.</ErrorMsg>
      </CircuitEquipQueryResponse>
   </s:Body>
</s:Envelope>
```

Call (Specifying Two Circuits)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:met="http://metegrity.com"
xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
   <soapenv:Header/>
   <soapenv:Body>
      <met:CircuitEquipQuery>
         <met:SiteName>[Visions SiteName]</met:SiteName>
         <met:UserName>[Visions UserName]</met:UserName>
         <met:Password>[Visions Password]</met:Password>
         <met:Values>
           [Records]
             [Values]
               [Value field="CircuitNum"]10-0006-X013[/Value]
             [/Values]
             [Values]
               [Value field="CircuitNum"]10-0006-X010[/Value]
             [/Values]
           [/Records]
         </met:Values>
      </met:CircuitEquipQuery>
   </soapenv:Body>
</soapenv:Envelope>
```

Call (Non-Circuit Input)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:met="http://metegrity.com"
xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
   <soapenv:Header/>
   <soapenv:Body>
      <met:CircuitEquipQuery>
         <met:SiteName>[Visions SiteName]</met:SiteName>
         <met:UserName>[Visions UserName]</met:UserName>
         <met:Password>[Visions Password]</met:Password>
         <met:CircuitNum>10-0006-X013-J02</met:CircuitNum>
         <met:PlantNum>1006</met:PlantNum>
         <met:EquipType>Pipe</met:EquipType>
      </met:CircuitEquipQuery>
   </soapenv:Body>
</soapenv:Envelope>
```

Response (Non-Circuit Input)

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Body>
      <CircuitEquipQueryResponse xmlns="http://metegrity.com">
         <CircuitEquipQueryResult>false</CircuitEquipQueryResult>
         <OutList i:nil="true"
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance"/>
         <moreLeft>false</moreLeft>
         <ErrorMsg>Specified equipment '1006/Pipe/10-0006-X013-J02' is not derived from
the Circuit base type; only Circuits are permitted.</ErrorMsg>
      </CircuitEquipQueryResponse>
   </s:Body>
</s:Envelope>
```

Additional Information

Visions Enterprise Help File

Configurations > Business Units

Configurations > Plants

Configurations > Type Customizations

Equipment Index

Lookup Data > F3 Lookup Lists

Static Data

Work Memos API

Overview

Several application program interfaces (API) are available to allow you to perform certain actions without using the Visions Enterprise user interface (UI). This allows third party developers to use web service calls to get or update Visions data.

The Visions API for work memos allows you to manage work memos, both inspection tasks and work requests; such as creating, updating, or querying work memo data.

Access

Navigation

The web service calls are expressed as a C# interface, with the individual calls, conventions, references, and examples documented below.

```
public interface IVisAPI {
...
}
```

Conditions

1. The Work Memo API is a separate module which must be active to use its calls.
2. Security to perform calls are dependent upon the security access for the username passed into a call. For example: if you do not have 'Add Inspection Tasks' or 'Add Work Requests' privileges, you will not be able to call the CreateWorkOrder call successfully. Security functions for Work Memos are:
   a. **Add Inspection Tasks**: controls creation of 'IT' work memos
   b. **Open/Edit Inspection Task**: controls open or editing of inspection task work memos
   c. **Delete Inspection Tasks**: controls deletion of inspection task work memos
   d. **Complete / Close Inspection Tasks**: controls closure or completion or inspection task work memos
   e. **Add Work Requests**: controls creation of 'WR' work memos
   f. **Open/Edit Work Request**: controls open or editing of work request work memos
   g. **Delete Work Requests**: controls deletion of work request work memos
   h. **Complete / Close Work Requests**: controls closure or completion or work request work memos

Input

The Work Memo API provides the following calls:

- **CreateWorkOrder**: create either an inspection task or work request; simulates the 'New' action on the Inspection Task Equipment File Log or the Work Request Equipment File Log

  Security: Permission is dependent upon the 'Add Inspection Task' and/or 'Add Work Request' functions dependent upon the data values within the call.

- **UpdateWorkOrder**: update an existing work memo

  Security: Permission is dependent upon the 'Open/Edit Inspection Task' and/or 'Open/Edit Work Request' functions dependent upon the data values within the call.

- **QueryWorkOrder**: query the work memos, returning the value(s) for the attribute(s) specified within the call

  Security: Permission is dependent upon the 'Open/Edit Inspection Task' and/or 'Open/Edit Work Request' functions dependent upon the data values within the call.

- **WorkOrderExistsQuery**: query the work memos; verifying whether the work memo exists

  Security: Permission is dependent upon the 'Open/Edit Inspection Task' and/or 'Open/Edit Work Request' functions dependent upon the data values within the call.

- **MemoListQuery**: query the work memos, returning the attributes from the Visions Work Memo Go Log for the specified memo numbers and types

  Security: Permission is dependent upon the 'Open/Edit Inspection Task' and/or 'Open/Edit Work Request' functions dependent upon the data values within the call.

- **WorkRequestListQuery**: query the work memos, returning the attributes from the Visions Work Memo Go Log for the specified memo numbers and 'WR' memo type

  Security: Permission is dependent upon the 'Open/Edit Work Request' function.

- **InspectionTaskListQuery**: query the work memos, returning the attributes from the Visions Work Memo Go Log for the specified memo numbers and 'IT' memo type

  Security: Permission is dependent upon the 'Open/Edit Inspection Task' function.

Calls

```
bool CreateWorkOrder(string SiteName, string UserName, string Password, string EquipNum,
   string PlantNum, string EquipType, string MemoType, string Values, out string MemoNo,
   out string ErrorMsg);

bool UpdateWorkOrder(string SiteName, string UserName, string Password, string MemoNo,
   string MemoType, string Values, out string ErrorMsg);

bool QueryWorkOrder(string SiteName, string UserName, string Password, string MemoNo,
   string MemoType, string FieldPath, out string FieldVal, out string ErrorMsg);

bool UpdateSchedule(string SiteName, string UserName, string Password, string EquipNum,
   string PlantNum, string EquipType, string DueDate, string DateFmt, out string
   ErrorMsg);
```

```
bool WorkOrderExistsQuery(string SiteName, string UserName, string Password, string
    MemoNo, string MemoType, out bool existsResult, out string ErrorMsg);

bool MemoListQuery(string SiteName, string UserName, string Password, string FieldName1,
    string[] FieldValues1, string FieldName2, string[] FieldValues2, string FieldName3,
    string[] FieldValues3, string FieldName4, string[] FieldValues4, string DateFmt,
    string ColumnSet, int pageSize, int pageNum,out string[][] OutList, out bool moreLeft,
    out string ErrorMsg);

bool WorkRequestListQuery(string SiteName, string UserName, string Password, string
    FieldName1, string[] FieldValues1, string FieldName2, string[] FieldValues2, string
    FieldName3, string[] FieldValues3, string FieldName4, string[] FieldValues4, string
    DateFmt, string ColumnSet, int pageSize, int pageNum, out string[][] OutList, out bool
    moreLeft, out string ErrorMsg);

bool InspectionTaskListQuery(string SiteName, string UserName, string Password, string
    FieldName1, string[] FieldValues1, string FieldName2, string[] FieldValues2, string
    FieldName3, string[] FieldValues3, string FieldName4, string[] FieldValues4, string
    DateFmt, string ColumnSet, int pageSize, int pageNum, out string[][] OutList, out bool
    moreLeft, out string ErrorMsg);
```

Calling Conventions

The web service is a standard SOAP web service and may be consumed by anything that can consume SOAP web services.

The following conventions are specific to the Work Memo API:

- API calls which reference a piece of equipment do so by its three (3) logical keys – PlantNum, EquipType, and EquipNum – not our internal Visions account identifier.
- API calls to interact with existing memos must reference them by the dual logical keys of MemoType and MemoNo, which together uniquely identify any work memo with values accessible to the user.
- When a new memo is created, it requires an equipment reference by logical keys and a memo type (either IT or WR), and returns the generated memo number (MemoNo).
- Arbitrary field values may be specified using the Values parameter, passing one or more field values as XML in a manner detailed below.
- The QueryWorkOrder calls return a single field value from the specified work memo as a string, which may be parsed as desired.
- The API web service ensures that the user account has the appropriate security permissions to perform the associated operation, (i.e., Open Inspection Tasks, Open Work Requests, Add Inspection Task, and Add Work Request).
- The WorkOrderExistsQuery has two boolean return values – the function returns false if the call fails for some reason (i.e., connectivity), but returns true and sets “existsResult” to true or false based on whether the item exists.

Behavior Notes

- The API web service will correct the casing of any values that already exist in the Visions database (i.e., MemoType, EquipNum, PlantNum, etc.) unless doing so would introduce ambiguity (i.e., conflicting records differing only in case exist), in which case it reads the values case-sensitively.
- The record creation API calls create a record in the associated parent table (Work), as well as zero or more child records in child tables (i.e, WLAttach, WLDistrib, etc.) needed to store values specified in the Values XML string.

List Parameters and Data Format

FieldValues: allow multiple list queries; each of which may take up to four fields in the form of a field name and list of permitted values

Note: All records matching any of the given values in each field are returned.

The list queries return two-dimensional string arrays which duplicate the grid indexes shown in the Visions Client. Row zero of the returned array has the column headers, and each row thereafter is a data row. By default, the returned columns directly mirror the relevant log in the Visions client, but they may be changed or limited by providing a ColumnSet.

ColumnSet: allows you to format the response dataset using the foreign references and field names to join related tables within a List Query call in an API, as well as assign an alias for your foreign reference(s); for example, to return only the logical keys for equipment, specify this ColumnSet value: “PlantNum,EquipNum,EquipType”

Reference the EquipmentListQuery call example for more information.

XML Field Values

The API calls accept values for arbitrary fields in a very simple XML format, illustrated below and in the examples section:

```
<Values>

<Value field='EquipName' >EX-01</Value>

<Value field='rfDesMat.MAWP' >560</Value>

<Value field='rfSchedule.DueDate'

 fmt='MM-dd-yyyy'>08-16-2012</Value>

</Values>
```

This could be sent to create a new piece of equipment with the specified Equipment Name, MAWP and Due Date.

The only significant tag is “Value”, and the only significant attributes are “field” and “fmt”. The field attribute is specified in the Tables and Fields group of References section below. This allows the specification of field values in child tables such as DesMat or Schedule when equipment is created or updated through the Asset API.

All date fields require that a format string be included (in the “fmt” attribute) with their value, so that we are able to interpret the value correctly and pre-emptively avoid any issues with date formatting, time zones, locale settings and so forth. The syntax of the date format string is the .NET standard, and is described here:

http://msdn.microsoft.com/en-us/library/az4se3k1

and here:

http://msdn.microsoft.com/en-us/library/8kb3ddd4

References

Data Model Overview

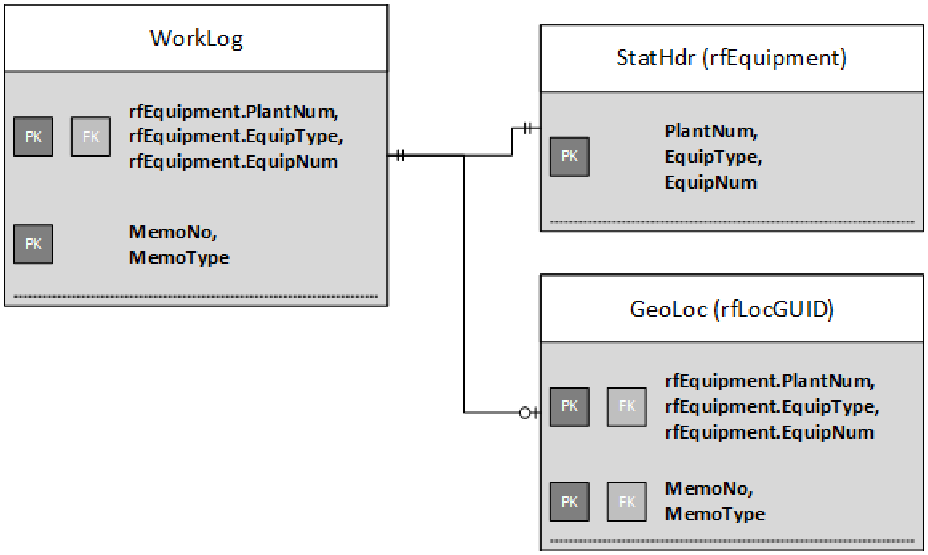


Table and Field Values

The table and field structures and definitions provide the ‘Values’ string within a call. Fields requiring the data to be entered based on a site configured unit of measure are identified by “Unit-Based Field”. Fields requiring the unit of measure value to be entered using the UoM Factors Lookup Data based on the Unit Category are identified by "UoM Factors Unit [Category]".

Limitation: Only the tables and fields listed below are supported for the Work Memo API.

StatHdr (rfEquipment)

Equipment identification within the **CreateWorkOrder** call.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**EquipNum**	("Equipment Number", dtString, Required Field, Read-Only Field, Maximum Length 35)
		Displays the asset or unique identification for the equipment.
string	**EquipType**	("Equipment Type", dtString, Required Field, Read-Only Field, Maximum Length 35, FK (EquipTypeList.EquipType))
		Displays the type of equipment.

string	**PlantNum**	("Plant ID", dtString, Required Field, Read-Only Field, Maximum Length 35, FK (PlantInfo.PlantNum))
		Displays the plant where the equipment is located.

Foreign References		
Table Name	**Call Value**	**Definition**
WorkLog	rfWorkLog	Master table for Work Memos

WorkLog

Work Memos: the main work memo entity which stores both inspection task and work request type memos.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
long	**MemoNo**	("Memo No", dtDouble, Required Field, Read-Only Field)
		A system generated memo number that uniquely identifies each work memo.
string	**MemoType**	("Type", dtString, Required Field, Maximum Length 3)
		Select the type of work memo; choices are: **IT** for Inspection Task or **WR** for Work Request.
string	AccessRequired	("Access Required", dtString, Maximum Length 20)
		From a lookup list, select the type of access required to perform the work for a work memo.
string	ActionToDate	("Action to Date", dtString, Maximum Length 4000)
		Enter the action to date for a work memo.
string	ActivityCode	("Activity Code", dtString, Maximum Length 5)
		From a lookup list, select the type of inspection activity for a work memo.
decimal?	ActualCost	("Actual Labour Cost", dtDouble, Unit-Based Field)
		Enter the actual cost of labour for a work memo.
int?	ActualHours	("Actual Hours", dtInteger)
		Enter the actual hours worked for a work memo.
decimal?	ActualMatCost	("Actual Mat. Cost", dtDouble, Unit-Based Field)
		Enter the actual cost of material for a work memo.
decimal?	AffectedProd	("Affected Production", dtDouble)
		The affected production volume for the work memo.
string	AffectedProdUnit	("Affected Prod Unit", dtString, Maximum Length 10, UoM Factors Unit [Capacity])
		From a lookup list, select the unit of measure for the affected production volume for the work memo.
bool?	Auditable	("Auditable", dtInteger)
		Indicate whether a work memo is auditable. Acceptable values: 0, N, No, 1, Y, Yes.
string	AuthInspAccept	("Post AI Approval", dtString, Maximum Length 35)

		Enter the name of the authorized inspector authorizing the post-approval sign-off. Limitation: The PostAIAcceptReqd field must be set to “yes” to populate a value.
string	AuthInspApprv	("Pre AI Approval", dtString, Maximum Length 35)
		Enter the name of the authorized inspector authorizing the pre-approval sign-off. Limitation: The PreAIAcceptReqd field must be set to “yes” to populate a value.
string	CleaningCode	("Cleaning Code", dtString, Maximum Length 5)
		From a lookup list, select the code identifying the type of cleaning required to perform the work for a work memo.
TG4?	CloseFlag	("Status", dtInteger, Required Field)
		Select the status of a work memo. Choices are: Open, Ready, Closed, or Completed. Limitations: When Closed is used as a value, values for the DateCompleted and CloseReason must be included. When Completed is used as a value, values for the DateCompleted, CompletedBy, and Completed must be included.
string	CloseReason	("Reason Closed", dtString, Maximum Length 50)
		Enter the reason a work memo was closed. Limitation: Only available when the CloseFlag is set to “Closed”.
string	CompCode	("Component Code", dtString, Maximum Length 5)
		From a lookup list, select the component code for the work performed in a work memo.
bool	Completed	("Completed", dtInteger, Required Field)
		Indicates whether a work memo is completed. Acceptable values: 0, N, No, 1, Y, Yes. Limitations: The DateCompleted and CompletedBy values must be populated. The CloseFlag value must be set to Completed. Once set to yes (true), a work memo cannot be changed to no (false).
string	CompletedBy	("Completed By", dtString, Maximum Length 20)
		The name of the person that completed the work memo; defaults to the current user and cannot be modified.
string	CreatedBy	("Created By", dtString, Maximum Length 35)
		The name of the user who created the work memo, defaults to the current user and cannot be modified.
DateTime?	DateCompleted	("Date Completed", dtDate)
		Enter the completion date for a work memo. If the memo extent is partial recurring, the memo will automatically be cloned. Limitation: When setting the DateCompleted, you must also set the CloseFlag to either Closed or Completed, and depending on the CloseFlag value other values may be required.
DateTime?	DateCreated	("Date Created", dtDate, Required Field, Read-Only Field)
		A system entered date that defaults to today.
DateTime?	DueDate	("Due Date", dtDate)
		Enter the due date for an inspection task.
bool?	EquipDown	("Equipment Down", dtInteger)

		Indicate whether the equipment is down for the work memo. Acceptable values: 0, N, No, 1, Y, Yes
bool?	ERPNotifReqd	("ERP Notification Required", dtInteger)
		Indicate whether an ERP notification is required for a work memo. Acceptable values: 0, N, No, 1, Y, Yes. Limitation: This is only available when an ERP module (SAP, EAM, JDE, Maximo) is active.
decimal?	EstimatedCost	("Estimated Cost", dtDouble, Unit-Based Field)
		Enter the estimated labor cost for the work performed in a work memo.
int?	EstimatedHours	("Estimated Hours", dtInteger)
		Enter the estimated hours for the work performed in a work memo.
decimal?	EstMatCost	("Estimated Material Cost", dtDouble, Unit-Based Field)
		Enter the estimated material cost for the work performed in a work memo.
TG5?	FullPartFlag	("Extent", dtInteger, Required Field)
		Select the extent of a work memo. Choices are: Full, Partial, and Partial-Recurring.
bool?	InslRemoval	("Insulation Removal", dtInteger)
		Indicate whether insulation removal is required to perform the work in a work memo. Acceptable values: 0, N, No, 1, Y, Yes.
string	InspectorName	("Inspector", dtString, Maximum Length 35)
		Enter the name of the inspector responsible for the work memo.
bool	IntegrityCritical	("Integrity Critical", dtInteger, Required Field)
		Indicate whether a work memo is integrity critical. Acceptable values: 0, N, No, 1, Y, Yes.
string	JobFileNo	("Job File Number", dtString, Maximum Length 35)
		Enter a job file number for a work number. Usually this references an external job file system.
bool?	LightingRequired	("Lighting Required", dtInteger)
		Indcate whether lighting is required to perform the work memo. Acceptable values: 0, N, No, 1, Y, Yes.
bool?	MajorWork	("Major Work", dtInteger)
		Indicate whether the work memo consists of major work. Acceptable values: 0, N, No, 1, Y, Yes.
string	MemoTitle	("Title", dtString, Maximum Length 100)
		Enter a title for a work memo. This will assist in sorting through work on the Work Memo logs. Note: User-configurable as a required field. Please check with your Visions Administrator.
bool?	MgmtOfChange	("Management of Change", dtInteger)
		Indicate whether management of change is required for a work memo. Acceptable values: 0, N, No, 1, Y, Yes.
string	NotificationType	("Notification Type", dtString, Maximum Length 5)

		From a lookup list, select the type of ERP notification that will be created or updated when the work memo is created or completed. Limitation: This is only available when an ERP module (SAP, EAM, JDE, Maximo) is active.
TG3?	OnOffFlag	("Stream", dtInteger)
		Select whether the work is done on or off stream and internally or externally. Choices are: Internal, External (On), or External (Off).
string	PaintCode	("Paint Code", dtString, Maximum Length 20)
		From a lookup list, select the paint code for the equipment for a work memo.
string	Permit	("Permit", dtString, Maximum Length 10)
		From a lookup list, select the type of permit required for a work memo.
bool?	PostAIAcceptReqd	("Post AI Required", dtInteger)
		Indicate whether the authorized inspector post-approval sign-off is required. Acceptable values: 0, N, No, 1, Y, Yes.
DateTime?	PostAISignoff	("Post AI Sign Off", dtDate)
		Select a date identifying when the authorized inspector signed off the post-approval of the work performed. Limitation: The PostAIAcceptReqd field must be set to "yes" to populate a value.
string	PostInspApprv	("Post Inspection Approval", dtString, Maximum Length 35)
		Enter the name of the inspector authorizing the post-approval sign-off. Limitation: The PostInspReqd field must be set to "yes" to populate a value.
bool?	PostInspReqd	("Post Inspection Required", dtInteger)
		Indicate whether the inspector post-approval sign-off is required. Acceptable values: 0, N, No, 1, Y, Yes.
DateTime?	PostInspSignoff	("Post Inspection Sign Off", dtDate)
		Select a date identifying when the inspector signed off the post-approval of the work performed. Limitation: The PostInspReqd field must be set to "yes" to populate a value.
string	PostMaintApprv	("Post Maintenance Approval", dtString, Maximum Length 35)
		Enter the name of the maintenance personnel authorizing the post-approval sign-off. Limitation: The PostMaintReqd field must be set to "yes" to populate a value.
bool?	PostMaintReqd	("Post Maintenance Required", dtInteger)
		Indicate whether the maintenance post-approval sign-off is required. Acceptable values: 0, N, No, 1, Y, Yes.
DateTime?	PostMaintSignoff	("Post Maintenance Sign Off", dtDate)
		Select a date identifying when the maintenance signed off the post-approval of the work performed. Limitation: The PostMaintReqd field must be set to "yes" to populate a value.
bool?	PreAIAcceptReqd	("Pre AI Required", dtInteger)
		Indicate whether the authorized inspector pre-approval sign-off is required. Acceptable values: 0, N, No, 1, Y, Yes.
DateTime?	PreAISignoff	("Pre AI Sign Off", dtDate)

		Select a date identifying when the authorized inspector signed off the pre-approval of the work to be performed. Limitation: The PreAIAcceptReqd field must be set to “yes” to populate a value.
string	PreEngApprv	("Pre Engineering Approval", dtString, Maximum Length 35)
		Enter the name of the engineering personnel authorizing the pre-approval sign-off. Limitation: The PreEngReqd field must be set to “yes” to populate a value.
bool?	PreEngReqd	("Pre Engineering Required", dtInteger)
		Indicate whether the engineering pre-approval sign-off is required. Acceptable values: 0, N, No, 1, Y, Yes.
DateTime?	PreEngSignoff	("Pre Engineering Sign Off", dtDate)
		Select a date identifying when engineering signed off the pre-approval of the work to be performed. Limitation: The PreEngReqd field must be set to “yes” to populate a value.
string	PreMaintApprv	("Pre Maintenance Approval", dtString, Maximum Length 35)
		Enter the name of the maintenance personnel authorizing the pre-approval sign-off. Limitation: The PreMaintReqd field must be set to “yes” to populate a value.
bool?	PreMaintReqd	("Pre Maintenance Required", dtInteger)
		Indicate whether the maintenance pre-approval sign-off is required. Acceptable values: 0, N, No, 1, Y, Yes.
DateTime?	PreMaintSignoff	("Pre Maintenance Sign Off", dtDate)
		Select a date identifying when maintenance signed-off the pre-approval of the work to be performed. Limitation: The PreMaintReqd field must be set to “yes” to populate a value.
string	PriorityCode	("Priority Code", dtString, Maximum Length 5)
		From a lookup list, select the code that identifies the priority for the work to be performed in a work memo.
string	ProbDesc	("Problem Desc", dtString, Maximum Length 4000)
		Enter a narrative description of the problem for a work memo.
string	ProblemCode	("Problem/Failure Code", dtString, Maximum Length 5)
		From a lookup list, select the code that identifies the problem or failure for the equipment for a work memo.
string	ProposedDisp	("Proposed Disp", dtString, Maximum Length 4000)
		Enter the proposed disposition for a work memo.
string	ReasonForWork	("Reason for Work", dtString, Maximum Length 25)
		From a lookup list, select the reason for work for a work memo.
string	RepairType	("Repair Type", dtString, Maximum Length 15)
		From a lookup list, select the type of repair performed on a work memo.
string	RFWCause	("RFW Cause", dtString, Maximum Length 25)
		From a lookup list, select the cause of the reason for work for a work memo.
int?	RInterval	("Interval", dtInteger, Unit-Based Field)

		Enter the partial recurring interval for a work memo. This is used in conjunction with the Date Completed to regenerate the work memo once it is completed. Limitation: Only available when the Extent is set to 'Partial-Recurring'.
bool?	Sentry	("Sentry", dtInteger) Identify whether a sentry is required for the work memo to be performed. Acceptable values: 0, N, No, 1, Y, Yes.
string	SurfacePrep	("Surface Preparation", dtString, Maximum Length 20) From a lookup list, select the type of surface preparation required for a work memo.
string	WorkDesc	("Work Desc", dtString, Maximum Length 4000) Enter a description of the work performed in the work memo.
string	WorkOrderNo	("Work Order No.", dtString, Maximum Length 35) Enter a workorder number for each work memo. Usually this references an external workorder system. Limitation: If the work memo is flagged as "ERP Required", the work order number is controlled by the ERP system and may not be populated through the calls.
string	WRDepartment	("Department", dtString, Maximum Length 25) From a lookup list, select the department that is to perform the work request.

Foreign References		
Table Name	**Call Value**	**Definition**
GeoLoc	rfLocGUID	Inspection Task / Work Request > Coordinates - both original GPS and Linear Note: Linear Coordinates are only available when the asset belongs to a Pipeline Plant.

GeoLoc (rfLocGUID)

Geolocation Coordinates: the centralized original GPS and linear coordinates for equipment, work, and trending.

Limitation: The fields listed below are based on the Plant to which the equipment belongs. If the plant is flagged as a Pipeline Plant, then the Linear coordinates are available; otherwise only the original GPS coordinates are available.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	Desc_End	("Description - End", dtString, Maximum Length 100) Enter a description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100) Enter a description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the ending milepost distance (chainage). The lookup list is extracted from the PL_SegMarkers Reference Markers.

		Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). The lookup list is extracted from the PL_SegMarkers Reference Markers. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field) The ending chainage or linear location. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field) The starting chainage or linear location. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger) From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage. Choices are: Left, Right. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger) From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage. Choices are: Left, Right. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field) The original ending GPS Longitude (X) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field) The original starting GPS Longitude (X) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Y_EndOrig	("Latitude - End", dtDouble, Unit-Based Field) The original ending GPS Latitude (Y) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field)

		The original starting GPS Latitude (Y) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field) The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - End", dtDouble, Unit-Based Field) The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign References		
Table Name	**Call Value**	**Definition**
WorkLog	lstWorkLog	Master table for Work Memos

Call Examples

Similar to the Asset API, a debug flag may be included in a call to echo the data in the response message for troubleshooting.

CreateWorkOrder

Call without Debug Information

```
<!-- This message uses CreateWorkOrder to create a sample Work Order -- a Work Request
    (WR) in this case -- for the D80-HC-22 Boiler. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:CreateWorkOrder>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>D80-HC-22</met:EquipNum>
    <met:PlantNum>U-3000 - Demonstration DB</met:PlantNum>
    <met:EquipType>Boiler</met:EquipType>
    <met:MemoType>WR</met:MemoType>
    <met:Values>
        [Values]
        [Value field='MemoTitle']VisAPI Test Work Memo[/Value]
        [Value field='ActualCost']77[/Value]
        [Value field='PaintCode']PC-2[/Value]
        [Value field='InspectorName']John Davlin[/Value]
        [/Values]
    </met:Values>
   </met:CreateWorkOrder>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The response message includes a MemoNo which can be stored and used to perform other
    operations on this work memo. -->
```

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <CreateWorkOrderResponse xmlns="http://metegrity.com">
    <CreateWorkOrderResult>true</CreateWorkOrderResult>
    <MemoNo>1625</MemoNo>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </CreateWorkOrderResponse>
 </s:Body>
</s:Envelope>
```

QueryWorkOrder

Call without Debug Information

```
<!-- Here, we use QueryWorkOrder to retrieve a specific field from the memo we created
    using CreateWorkOrder. Note that you have to specify both the MemoType (WR) and
    MemoNo (1625), as MemoNo values are only unique relative to Work Requests or
    Inspection Tasks. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:QueryWorkOrder>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:MemoNo>1625</met:MemoNo>
    <met:MemoType>WR</met:MemoType>
    <met:FieldPath>PaintCode</met:FieldPath>
   </met:QueryWorkOrder>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The Response returns a value for the requested field. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <QueryWorkOrderResponse xmlns="http://metegrity.com">
    <QueryWorkOrderResult>true</QueryWorkOrderResult>
    <FieldVal>PC-2</FieldVal>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </QueryWorkOrderResponse>
 </s:Body>
</s:Envelope>
```

UpdateWorkOrder

Call without Debug Information

```
<!-- Here we use UpdateWorkOrder to complete our sample memo. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:UpdateWorkOrder>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:MemoNo>1625</met:MemoNo>
    <met:MemoType>WR</met:MemoType>
    <met:Values>
        [Values]
          [Value field='CloseFlag']Completed[/Value]
          [Value field='Completed']true[/Value]
          [Value field='CloseReason']Third-Party API Integration[/Value]
          [Value field='DateCompleted' fmt='yyyyMMdd']20160305[/Value]
        [/Values>
    </met:Values>
   </met:UpdateWorkOrder>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The Response indicates the update succeeded. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <UpdateWorkOrderResponse xmlns="http://metegrity.com">
    <UpdateWorkOrderResult>true</UpdateWorkOrderResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </UpdateWorkOrderResponse>
 </s:Body>
</s:Envelope>
```

WorkOrderExistsQuery

Call without Debug Information

```
<!-- This simple call verifies that a Work Memo (identified by both
 MemoType and MemoNo in conjunct) exists in the database. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:WorkOrderExistsQuery>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:MemoNo>1625</met:MemoNo>
    <met:MemoType>WR</met:MemoType>
   </met:WorkOrderExistsQuery>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- Yes, it does. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <WorkOrderExistsQueryResponse xmlns="http://metegrity.com">
    <WorkOrderExistsQueryResult>true</WorkOrderExistsQueryResult>
    <existsResult>true</existsResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </WorkOrderExistsQueryResponse>
 </s:Body>
</s:Envelope>
```

MemoListQuery

The WorkRequestListQuery and InspectionTaskListQuery calls simply point to MemoListQuery which implicitly restricts the selection to Memo Type = ’IT’ or Memo Type = ’WR’, depending on which call is used so that you do not have to include the MemoType within the values.

Call without Debug Information

```
<!-- Here we use MemoListQuery to retrieve all of the Work Requests with
 MemoNo 1620-1625. Unlike with the EquipmentListQuery example, here
 we leave ColumnSet null (so we get the standard Visions Log fields)
 and we leave the pageSize/pageNum fields null (so everything is
 returned in a single Response message. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com"
   xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
 <soapenv:Header/>
 <soapenv:Body>
   <met:MemoListQuery>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:FieldName1>MemoNo</met:FieldName1>
    <met:FieldValues1>
      <arr:string>1620</arr:string>
      <arr:string>1621</arr:string>
      <arr:string>1622</arr:string>
      <arr:string>1623</arr:string>
      <arr:string>1624</arr:string>
      <arr:string>1625</arr:string>
    </met:FieldValues1>
    <met:FieldName2>MemoType</met:FieldName2>
    <met:FieldValues2>
      <arr:string>WR</arr:string>
    </met:FieldValues2>
    <!--
    <met:pageSize>?</met:pageSize>
    <met:pageNum>?</met:pageNum>
    -->
   </met:MemoListQuery>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The Response includes the requested data in a two-dimensional
 OutList array, with OutList[0] containing the Column Header
 names. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <MemoListQueryResponse xmlns="http://metegrity.com">
    <MemoListQueryResult>true</MemoListQueryResult>
    <OutList xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
      xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
      <a:ArrayOfstring>
       <a:string>Plant ID</a:string>
       <a:string>Equipment Type</a:string>
       <a:string>Equipment Number</a:string>
```

```
        <a:string>Associated Major</a:string>
        <a:string>Type</a:string>
        <a:string>Memo No</a:string>
        <a:string>Date Created</a:string>
        <a:string>Title</a:string>
        <a:string>Completed</a:string>
        <a:string>Status</a:string>
        <a:string>Problem/Failure Code</a:string>
        <a:string>Activity Code</a:string>
        <a:string>Extent</a:string>
        <a:string>Stream</a:string>
        <a:string>Interval</a:string>
        <a:string>Job File Number</a:string>
        <a:string>Work Order No.</a:string>
        <a:string>Priority Code</a:string>
        <a:string>Date Completed</a:string>
        <a:string>Reason Closed</a:string>
        <a:string>Due Date</a:string>
        <a:string>WE Name</a:string>
        <a:string>WT Number</a:string>
        <a:string>Assigned User</a:string>
       </a:ArrayOfstring>
       <a:ArrayOfstring>
        <a:string>U-3000 - Demonstration DB</a:string>
        <a:string>Vessel</a:string>
        <a:string>D08-V-100</a:string>
        <a:string i:nil="true"/>
        <a:string>WR</a:string>
        <a:string>1624</a:string>
        <a:string>26/01/2016</a:string>
        <a:string>testing the component control</a:string>
        <a:string>False</a:string>
        <a:string>Ready</a:string>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string>Partial</a:string>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
       </a:ArrayOfstring>
       <a:ArrayOfstring>
        <a:string>JK-Plant0</a:string>
        <a:string>Air Cooler</a:string>
        <a:string>1</a:string>
        <a:string i:nil="true"/>
        <a:string>IT</a:string>
        <a:string>1622</a:string>
        <a:string>11/01/2016</a:string>
        <a:string i:nil="true"/>
        <a:string>False</a:string>
        <a:string>Open</a:string>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string>0</a:string>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
       </a:ArrayOfstring>
   </OutList>
    <moreLeft>false</moreLeft>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </MemoListQueryResponse>
 </s:Body>
</s:Envelope>
```

Additional Information

Visions Enterprise Help File

Lookup Data > F3 Lookup Lists

Work Cycle > Work Memos

Visions Administrator

Settings > Required Fields

Integrity Reports API

Overview

Several application program interfaces (API) are available to allow you to perform certain actions without using the Visions Enterprise user interface (UI). This allows third party developers to use web service calls to get or update Visions data.

The Visions API for fixed integrity reports allows you to manage fixed integrity reports; such as creating, updating, or querying fixed integrity report data.

Access

Navigation

The web service calls are expressed as a C# interface, with the individual calls, conventions, references, and examples documented below.

```
public interface IVisAPI {
...
}
```

Conditions

1. The IR API is a separate module which must be active to use its calls.
2. Security to perform calls are dependent upon the security access for the username passed into a call. For example: if you do not have 'Add Fixed IR' privileges, you will not be able to call the CreateFixedIR call successfully. Security functions for Fixed Integrity Reports are:
    a. **Add Fixed IR**: controls creation of 'IR' integrity reports
    b. **Open/Edit Fixed IR**: controls open or editing of fixed integrity reports
3. The IR API does not include the ability to:
    a. complete or close an integrity report
    b. create details for minor equipment, circuited equipment, or components of the integrity report's equipment
    c. create attachments for an integrity report
    d. link check lists and respond to questions
    e. link additional external comments
    f. create additional actions
    g. create activity codes
    h. create or link anomalies
    i. create repairs

j. create and link recommendations

k. link distribution names

l. link additional reports; such as NDE reports, work memos, or references

m. delete an integrity report

4. Fixed Integrity Reports are limited to certain types of equipment. They cannot be created for types derived from the following supplied types:

   a. Centrifugal Pump

   b. GP Steam Turbine

   c. RE Motor

   d. PSV

   e. PSV Location

Input

The Integrity Report API provides the following calls:

◢ **CreateIntegrityReport**: create a fixed integrity report; simulates the 'New' action on the Fixed Integrity Report Equipment File Log.

When a new integrity report is created, the Detail record will be created using the integrity report's owning asset.

When a new integrity report is created, the External Comments in the Details will be automatically created using records in the External Comments Lookup Data that are flagged as 'IR Default' and are assigned to the same equipment type as the integrity report's owning asset.

When a new integrity report is created, the Actions in the Details will be automatically created using records in the IR Actions Lookup Data that are flagged as 'IR Default'.

Security: Permission is dependent upon the 'Add Fixed IR' function dependent upon the data values within the call.

◢ **UpdateIntegrityReport**: update an existing fixed integrity report

Security: Permission is dependent upon the 'Open/Edit Fixed IR' function dependent upon the data values within the call.

◢ **QueryIntegrityReport**: query the fixed integrity reports, returning the value(s) for the attribute(s) specified within the call

Security: Permission is dependent upon the 'Open/Edit Fixed IR' function dependent upon the data values within the call.

◢ **IntegrityReportExistsQuery**: query the fixed integrity reports; verifying whether the fixed integrity report exists

Security: Permission is dependent upon the 'Open/Edit Fixed IR' function dependent upon the data values within the call.

- **IntegrityReportListQuery**: query the fixed integrity reports, returning the attributes from the Visions Fixed integrity report Go Log for the specified memo numbers and types

  Security: Permission is dependent upon the 'Open/Edit Fixed IR' function dependent upon the data values within the call.

Calls

```
bool CreateIntegrityReport (string SiteName, string UserName, string Password, string
    EquipNum, string PlantNum, string EquipType, string MemoType, string Values, out string
    MemoNo, out string ErrorMsg, string MapID);

bool UpdateIntegrityReport (string SiteName, string UserName, string Password, string
    MemoNo, string MemoType, string Values, out string ErrorMsg, string MapID);

bool QueryIntegrityReport (string SiteName, string UserName, string Password, string
    MemoNo, string MemoType, string FieldPath, out string FieldVal, out string ErrorMsg,
    string MapID, string Values);

bool IntegrityReportExistsQuery (string SiteName, string UserName, string Password, string
    MemoNo, string MemoType, out bool existsResult, out string ErrorMsg, string MapID,
    string Values);

bool IntegrityReportListQuery (string SiteName, string UserName, string Password, string
    FieldName1, string[] FieldValues1, string FieldName2, string[] FieldValues2, string
    FieldName3, string[] FieldValues3, string FieldName4, string[] FieldValues4, string
    DateFmt, string ColumnSet, int pageSize, int pageNum, out string[][] OutList, out bool
    moreLeft, out string ErrorMsg, string MapID);
```

Calling Conventions

The web service is a standard SOAP web service and may be consumed by anything that can consume SOAP web services.

The following conventions are specific to the Fixed Integrity Report API:

- API calls which reference a piece of equipment do so by its three (3) logical keys – PlantNum, EquipType, and EquipNum – not our internal Visions account identifier.
- API calls to interact with existing memos must reference them by the dual logical keys of MemoType and MemoNo, which together uniquely identify any fixed integrity report with values accessible to the user.
- When a new integrity report is created, it requires an equipment reference by logical keys and a memo type (IR), and returns the generated memo number (MemoNo).
- Arbitrary field values may be specified using the Values parameter, passing one or more field values as XML in a manner detailed below.
- The QueryIntegrityReport call return a single field value from the specified fixed integrity report as a string, which may be parsed as desired.
- The API web service ensures that the user account has the appropriate security permissions to perform the associated operation, (i.e., Open/Edit Fixed IR, Add Fixed IR).

- The IntegrityReportExistsQuery has two boolean return values – the function returns false if the call fails for some reason (i.e., connectivity), but returns true and sets “existsResult” to true or false based on whether the item exists.

Behavior Notes

- The API web service will correct the casing of any values that already exist in the Visions database (i.e., MemoType, EquipNum, PlantNum, etc.) unless doing so would introduce ambiguity (i.e., conflicting records differing only in case exist), in which case it reads the values case-sensitively.
- The record creation API call creates a record in the associated parent table (Fixed IR), as well as one child record in the child table (i.e., Fixed IR Details, etc.) needed to store values specified in the Values XML string.

List Parameters and Data Format

FieldValues: allow multiple list queries; each of which may take up to four fields in the form of a field name and list of permitted values

Note: All records matching any of the given values in each field are returned.

The list queries return two-dimensional string arrays which duplicate the grid indexes shown in the Visions Client. Row zero of the returned array has the column headers, and each row thereafter is a data row. By default, the returned columns directly mirror the relevant log in the Visions client, but they may be changed or limited by providing a ColumnSet.

ColumnSet: allows you to format the response dataset using the foreign references and field namesto join related tables within a List Query call in an API, as well as assign an alias for your foreign reference(s); for example, to return only the logical keys for equipment, specify this ColumnSet value: “PlantNum,EquipNum,EquipType”

Reference the EquipmentListQuery call example for more information.

XML Field Values

The API calls accept values for arbitrary fields in a very simple XML format, illustrated below and in the examples section:

```
<Values>

<Value field='EquipName' >EX-01</Value>

<Value field='rfDesMat.MAWP' >560</Value>

<Value field='rfSchedule.DueDate'

 fmt='MM-dd-yyyy'>08-16-2012</Value>

</Values>
```

This could be sent to create a new piece of equipment with the specified Equipment Name, MAWP and Due Date.

The only significant tag is “Value”, and the only significant attributes are “field” and “fmt”. The field attribute is specified in the Tables and Fields group of References section below. This allows the specification of field values in child tables such as DesMat or Schedule when equipment is created or updated through the Asset API.

All date fields require that a format string be included (in the "fmt" attribute) with their value, so that we are able to interpret the value correctly and pre-emptively avoid any issues with date formatting, time zones, locale settings and so forth. The syntax of the date format string is the .NET standard, and is described in the following sites:

https://docs.microsoft.com/en-us/dotnet/standard/base-types/standard-date-and-time-format-strings

https://docs.microsoft.com/en-us/dotnet/standard/base-types/custom-date-and-time-format-strings

References

Data Model Overview

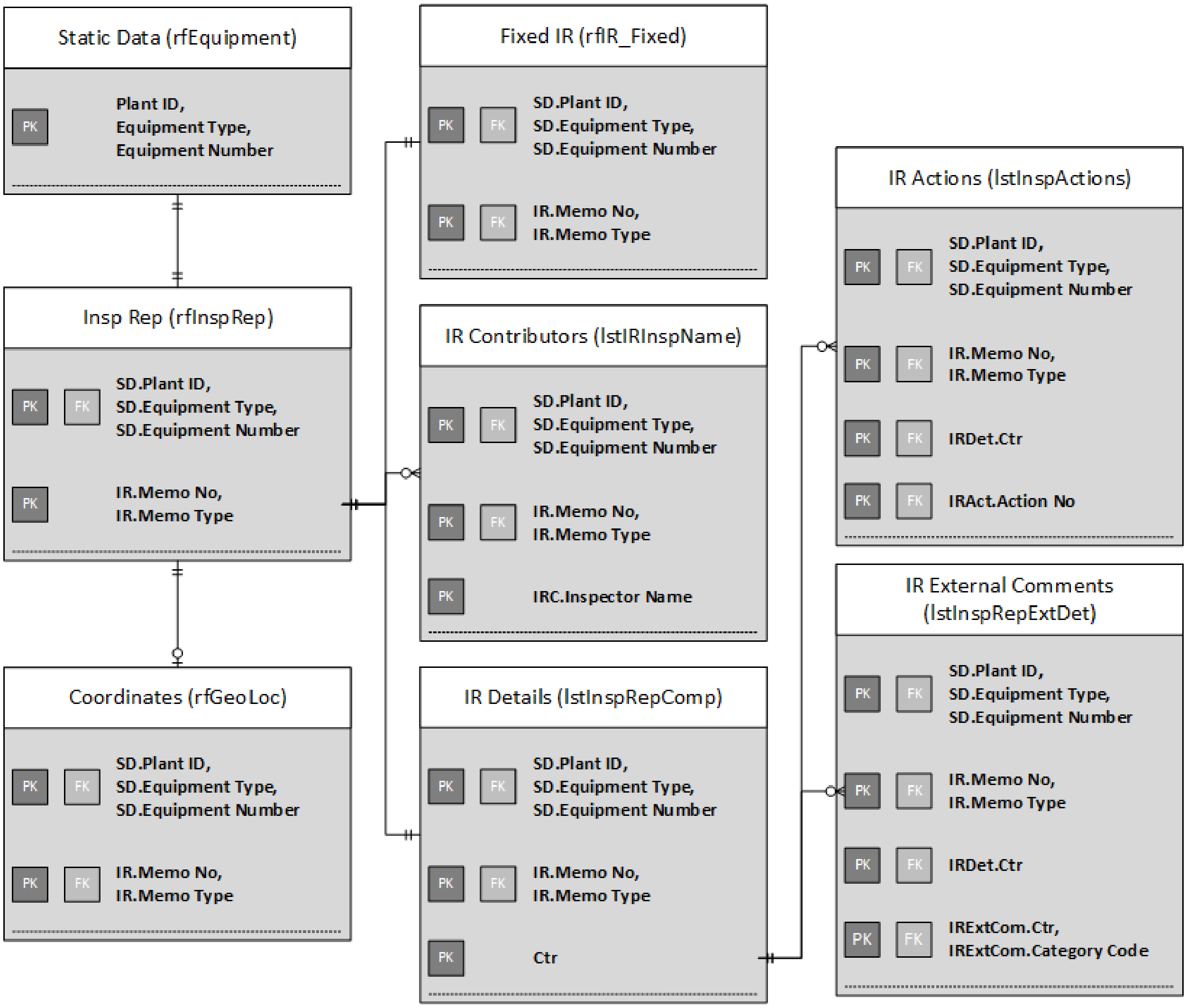


Table and Field Values

The table and field structures and definitions provide the 'Values' string within a call. Fields requiring the data to be entered based on a site configured unit of measure are identified by "Unit-Based Field".

Limitation: Only the tables and fields listed below are supported for the Integrity Report API.

StatHdr (rfEquipment)

Equipment identification within the calls.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**PlantNum**	("Plant ID", dtString, Required Field, Read-Only Field, Maximum Length 35, FK (PlantInfo.PlantNum))
		Displays the plant where the equipment is located.
string	**EquipType**	("Equipment Type", dtString, Required Field, Read-Only Field, Maximum Length 35, FK (EquipTypeList.EquipType))
		Displays the type of equipment.
string	**EquipNum**	("Equipment Number", dtString, Required Field, Read-Only Field, Maximum Length 35)
		Displays the asset or unique identification for the equipment.

Foreign References		
Table Name	**Call Value**	**Definition**
InspRep	lstInspRep	Master table for Integrity Reports

Integrity Report (rfInspRep)

Integrity Report: the main fixed integrity report entity which stores both fixed and rotating integrity report type memos.

Limitation: Currently only Fixed Integrity Report memo types (IR) are supported in the API.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
long	**MemoNo**	("Memo No", dtDouble, Required Field, Read-Only Field)
		A system generated memo number that uniquely identifies each fixed integrity report.
string	**MemoType**	("Type", dtString, Required Field, Maximum Length 3)
		A system value that identifies it is an integrity report. Choice is: **IR** for Fixed Integrity Report.
string	CreatedBy	("Created By", dtString, Read-Only Field, Maximum Length 35)
		The name of the user who created the integrity report, defaults to the current user and cannot be modified.
DateTime?	DateCreated	("Date Created", dtDate, Required Field, Read-Only Field)
		A system entered date that defaults to today on creation of an integrity report.
TG5?	FullPartFlag	("Extent", dtInteger)

		Select the extent of an integrity report. Choices are: Full, Partial. Note: User-configurable with a default value or as a required field. Please check with your Visions Administrator.
DateTime?	InspDate	("Date Inspected", dtDate)
		The date that the inspection was performed or finished.
string	InspSummary	("Executive Summary", dtString, Maximum Length 4000)
		A brief summary of the inspection scope and results. It is primarily intended to be used as an Executive Summary; an overview of what was done and what was found.
bool?	IntExtFlag	("Visual External", dtInteger)
		Indicate whether the integrity report includes a visual external inspection. Acceptable values: 0, N, No, 1, Y, Yes. Note: User-configurable as a required field. Please check with your Visions Administrator.
string	JobFileNo	("Job File Number", dtString, Maximum Length 35)
		Enter a job file number for a work number. Usually this references an external job file system. Note: User-configurable as a required field. Please check with your Visions Administrator.
string	MemoTitle	("Title", dtString, Maximum Length 100)
		Enter a descriptive title for an integrity report. This will assist in sorting through work on the Fixed IR logs. Note: User-configurable as a required field. Please check with your Visions Administrator.
TG3?	OnOffFlag	("Stream", dtInteger)
		Select whether the work is done on or off stream and internally or externally. Choices are: Internal, External (On), or External (Off). Note: User-configurable as a required field. Please check with your Visions Administrator.
string	Reason	("Reason", dtString, F3 Lookup List, Maximum Length 50)
		From a lookup list, select the reason for inspection for the inspection report. Note: User-configurable as a required field. Please check with your Visions Administrator.
string	ReportNumber	("Report Number", dtString, Maximum Length 35)
		An identifying report number for the inspection; typically, a corporate numbering standard is used or a reference to an external reporting system. Note: User-configurable as a required field. Please check with your Visions Administrator.
string	WorkOrderNo	("Work Order No.", dtString, Maximum Length 35)
		Enter a work order number for each integrity report. Usually this references an external work order system. Limitation: If the Fixed Integrity Report is linked to an ERP system, this field will become read-only and cannot be updated. Note: User-configurable as a required field. Please check with your Visions Administrator.

Foreign References		
Table Name	**Call Value**	**Definition**
GeoLoc	rfLocGUID	Fixed IR > Coordinates - both original GPS and Linear Note: Linear Coordinates are only available when the asset belongs to a Pipeline Plant.
InspRepComp	lstInspRepComp	Fixed IR > Details - contains the inspection details, actions, external comments, etc.
IR_Fixed	rfIR_Fixed	Fixed IR > Summary - the individual fields specific to a Fixed Integrity Report
IRInspName	lstIRInspName	Fixed IR > Summary > Contributors - the inspectors who performed or contributed to the inspection
StatHdr	rfEquipment	Fixed IR > Equipment information

Fixed IR Detail (rfIR_Fixed)

Fixed Integrity Report detail: the integrity report details specifically for the fixed integrity reports; Summary and Process History.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
long	**MemoNo**	("Memo No", dtInteger, Required Field, Read-Only Field) A system generated memo number that uniquely identifies each fixed integrity report.
string	**MemoType**	("Type", dtString, Required Field, Maximum Length 3) Select the type of fixed integrity report; choice is: **IR** for Fixed Integrity Report.
string	AuthInspName	("A I Name", dtString, Maximum Length 35) The name of the authorized inspector for the inspection.
string	ChiefInsp	("Chief Inspector", dtString, Maximum Length 35) The name of the chief inspector responsible for the inspection.
DateTime?	ChiefInspDate	("Chief Inspector Sign-off", dtDate) The date the chief inspector signed off the inspection.
decimal?	OperTime	("Operating Time", dtDouble, Unit-Based Field) The amount of time, typically the number of hours, the equipment has been in operation since the last inspection.
string	Process	("Process", dtString, Maximum Length 4000) The process conditions (operating temperatures, levels, pressures, fluids, etc.) and information you would want to ensure personnel performing inspections in the future would know about.
string	ProcessCategory	("Process Category", dtString, F3 Lookup, Maximum Length 20) From a lookup list, select a category for the process / continuous history. This will help categorize the continuous history summary.
string	RegulatoryComment	("Regulatory Comment", dtString, Maximum Length 100) A short, descriptive comment for the Regulatory board; usually describing the state of the equipment found during inspection.

bool	ResetOperTime	("Reset Time", dtInteger, Required Field) Indicate whether the accumulative total operating time is to be reset and start from this inspection forward. Typically, when major work is performed or the equipment is rerated. Choices are: 0, N, No, 1, Y, Yes. **Note**: Defaults to "N" (No, 0) unless otherwise specified.

Foreign References		
Table Name	**Call Value**	**Definition**
InspRep	rfInspRep	Master table for Integrity Reports

IR Contributors (lstIRInspName)

Integrity Report Contributors: the list of inspectors that contribute to the integrity report.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
long	**MemoNo**	("Memo No", dtDouble, Required Field, Read-Only Field) A system generated memo number that uniquely identifies each fixed integrity report.
string	**MemoType**	("Type", dtString, Required Field, Maximum Length 3) Set the type of fixed integrity report; choice is: **IR** for Fixed Integrity Report.
string	**InspectorName**	("Inspector Name", dtString, Required Field, Maximum Length 35) The name of the inspector that performed or contributed to the inspection.
string	Department	("Department", dtString, F3 Lookup, Maximum Length 25) From a lookup list, select the department for the contributing inspector or analyst.

Foreign References		
Table Name	**Call Value**	**Definition**
InspRep	rfInspRep	Master table for Integrity Reports

IR Details (lstInspRepComp)

Integrity Report Details: the fixed integrity report detail entity which stores the inspection details, including actions and external comments, for the owning asset.

Limitation: Currently only Fixed Integrity Report Details for the owning asset can be created in the API. Integrity report details cannot be created for circuited equipment when the owning asset is classified as Circuit, minor equipment when the owning asset is classified as Major, or components of the owning asset.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
long	**MemoNo**	("Memo No", dtDouble, Required Field, Read-Only Field)

		A system generated memo number that uniquely identifies each fixed integrity report.
string	**MemoType**	("Type", dtString, Required Field, Maximum Length 3) Set the type of fixed integrity report; choices are: **IR** for Fixed Integrity Report.
long	**IRCompCounter**	("IR Comp Counter", dtInteger, Required Field, Read-Only Field) A system counter that uniquely identifies each selected inspection report detail; the selected equipment, minor equipment, circuited equipment.
string	InspDetails	("IR Details", dtString, Maximum Length 4000) A descriptive, detailed explanation of the inspection work performed.
string	ProblemCode	("Problem Code", dtString, F3 Lookup, Maximum Length 5) From a lookup list, select the code that identifies the problem or failure for an inspection detail.
bool	Completed	("Detail Completed", dtInteger, Required Field) Indicates whether the inspection for the detail record is complete. Choices are: 0, N, No, 1, Y, Yes. Note: Defaults to “N” (No, 0) unless otherwise specified.

Foreign References		
Table Name	**Call Value**	**Definition**
InspActions	lstInspActions	Integrity Report > Details > Actions
InspRepExtDet	lstInspRepExtDet	Integrity Report > Details > External Comments
InspRep	rfInspRep	Master table for Integrity Reports

IR Details - Actions (lstInspActions)

Integrity Report Detail Actions: the list of actions for each integrity report detail.

Note: Actions are automatically added on creation of the Fixed IR Detail record from the IR Actions Lookup table for the IR Default options that are checked (Yes).

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
long	**MemoNo**	("Memo No", dtDouble, Required Field, Read-Only Field) A system generated memo number that uniquely identifies each fixed integrity report.
string	**MemoType**	("Type", dtString, Required Field, Maximum Length 3) Select the type of fixed integrity report; choice is: **IR** for Fixed Integrity Report.
long	**ActionNum**	("Action No", dtInteger, Required Field, Read-Only Field) A system counter that uniquely identifies each integrity report action.
string	Category	("Category", dtString, F3 Lookup List, Maximum Length 30) From a lookup list, select the category of the action.
string	Action	("Action", dtString, Maximum Length 255) Select or enter an action for an inspection report.
bool	Completed	("Detail Completed", dtInteger, Required Field)

		Indicates whether the action for the inspection of the detail record is complete. Choices are: 0, N, No, 1, Y, Yes. **Note**: Defaults to “N” (No, 0) unless otherwise specified.
string	ActionTaken	("Action", dtString, Maximum Length 4000)
		Enter any action(s) taken for the individual action record during the inspection.

Foreign References		
Table Name	**Call Value**	**Definition**
InspRepComp	rfInspRepComp	Fixed IR > Details

IR Details - External Comments (lstInspRepExtDet)

Integrity Report Detail External Comments: the list of external comments for each integrity report detail.

Note: External Comments are automatically added on creation of the Fixed IR Detail record from the External Comments Lookup table for the IR Default options that are checked (Yes) and the External Comments Equipment Type matches the equipment type derived from the owning asset.

Limitation: Currently additional external comments cannot be linked through the API, they must be linked through the Visions Enterprise Client application. Once added, the External Detail and IR Comments may be updated.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
long	**MemoNo**	("Memo No", dtDouble, Required Field, Read-Only Field)
		A system generated memo number that uniquely identifies each fixed integrity report.
string	**MemoType**	("Type", dtString, Required Field, Maximum Length 3)
		Select the type of fixed integrity report; choice is: **IR** for Fixed Integrity Report.
long	**IRExtDetCtr**	("IR Ext Detail Counter", dtInteger, Required Field, Read-Only Field)
		A system counter that uniquely identifies each external detail in an Inspection Report.
string	**CategoryCode**	("IR Ext Detail Counter", dtString, Required Field, Maximum Length 30)
		Identifies the external detail category, selected from the external details list dialog.
string	**Comments**	("External Detail", dtString, Required Field, Maximum Length 255)
		Insert an appropriate external detail for an Inspection Report.
string	IRComments	("IR Comment", dtString, Maximum Length 255)
		Enter any additional comments for the External Detail.

Foreign References		
Table Name	**Call Value**	**Definition**
InspRepComp	rfInspRepComp	Fixed IR > Details

GeoLoc (rfLocGUID)

Geolocation Coordinates: the centralized original GPS and linear coordinates for the integrity report.

Limitation: The fields listed below are based on the Plant to which the equipment belongs. If the plant is flagged as a Pipeline Plant, then the Linear coordinates are available; otherwise only the original GPS coordinates are available.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	Desc_End	("Description - End", dtString, Maximum Length 100) Enter a description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100) Enter a description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the ending milepost distance (chainage). The lookup list is extracted from the PL_SegMarkers Reference Markers. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). The lookup list is extracted from the PL_SegMarkers Reference Markers. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field) The ending chainage or linear location. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field) The starting chainage or linear location. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger) From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage. Choices are: Left, Right. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger) From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage. Choices are: Left, Right. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.

decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field) The original ending GPS Longitude (X) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field) The original starting GPS Longitude (X) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Y_EndOrig	("Latitude - End", dtDouble, Unit-Based Field) The original ending GPS Latitude (Y) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field) The original starting GPS Latitude (Y) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field) The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - End", dtDouble, Unit-Based Field) The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign References		
Table Name	**Call Value**	**Definition**
InspRep	lstInspRep	Master table for Integrity Reports

Call Examples

Similar to the Asset API, a debug flag may be included in a call to echo the data in the response message for troubleshooting.

CreateIntegrityReport

Call without Debug Information

```
<!-- This message uses CreateIntegrityReportto create a sample Integrity Report -- a
    Fixed Integrity Report (IR) in this case -- for the 01-V-100 Vessel. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:met="http://metegrity.com">
   <soapenv:Header/>
   <soapenv:Body>
      <met:CreateIntegrityReport>
```

```
         <met:SiteName>[Visions SiteName]</met:SiteName>
         <met:UserName>[Visions UserName]</met:UserName>
         <met:Password>[Visions Password]</met:Password>
         <met:EquipNum>01-V-100</met:EquipNum>
         <met:PlantNum>Plant 01</met:PlantNum>
         <met:EquipType>Vessel</met:EquipType>
         <met:MemoType>IR</met:MemoType>
         <met:Values>
           [Values]
             [Value field='MemoTitle']IR API Testing[/Value]
             [Value field='FullPartFlag']Partial[/Value]
             [Value field='OnOffFlag']External (On)[/Value]
             [Value field='ReportNumber']77[/Value]
             [Value field='rfIR_Fixed.Process']Test Process Value[/Value]
             [Value field='lstInspRepComp.InspDetails']Details Text One[/Value]
             [Value field='lstIRInspName[1].InspectorName']Julian[/Value]
             [Value field='lstIRInspName[1].Department']Operations[/Value]
             [Value field='lstIRInspName[2].InspectorName']Dawid[/Value]
             [Value field='lstIRInspName[2].Department']Inspection[/Value]
             [Value field='lstIRInspName[3].InspectorName']Carolyn[/Value]
             [Value field='lstIRInspName[3].Department']Maintenance[/Value]
           [/Values]
         </met:Values>
      </met:CreateIntegrityReport>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The response message includes a MemoNo which can be stored and used to perform other
    operations on this fixed integrity report. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Body>
      <CreateIntegrityReportResponse xmlns="http://metegrity.com">
         <CreateIntegrityReportResult>true</CreateIntegrityReportResult>
         <MemoNo>1414</MemoNo>
         <ErrorMsg>Operation successful.</ErrorMsg>
      </CreateIntegrityReportResponse>
  </s:Body>
</s:Envelope>
```

QueryIntegrityReport

Call without Debug Information

```
<!-- Here, we use QueryIntegrityReport to retrieve a specific field from the integrity
    report we created using CreateIntegrityReport. Note that you have to specify both the
    MemoType (IR) and MemoNo (1414), as MemoNo values are only unique relative to Fixed
    Integrity Reports. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:QueryWorkOrder>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:MemoNo>1625</met:MemoNo>
    <met:MemoType>WR</met:MemoType>
    <met:FieldPath>PaintCode</met:FieldPath>
   </met:QueryWorkOrder>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The Response returns a value for the requested field. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Body>
      <QueryIntegrityReportResponse xmlns="http://metegrity.com">
         <QueryIntegrityReportResult>true</QueryIntegrityReportResult>
         <FieldVal>Partial visual inspection.</FieldVal>
         <ErrorMsg>Operation successful.</ErrorMsg>
      </QueryIntegrityReportResponse>
   </s:Body>
</s:Envelope>
```

UpdateIntegrityReport

Call without Debug Information

```
<!-- Here we use UpdateIntegrityReport to update the Inspection Details and a couple of
    fields. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:met="http://metegrity.com">
   <soapenv:Header/>
   <soapenv:Body>
      <met:UpdateIntegrityReport>
         <met:SiteName>[Visions SiteName]</met:SiteName>
         <met:UserName>[Visions UserName]</met:UserName>
         <met:Password>[Visions Password]</met:Password>
         <met:MemoNo>1345</met:MemoNo>
         <met:MemoType>IR</met:MemoType>
         <met:Values>
           [Values]
             [Value field='FullPartFlag']Partial[/Value]
             [Value field='rfIR_Fixed.Process']Test Process Value[/Value]
             [Value field='lstInspRepComp[1].InspDetails']Details Text One[/Value]
           [/Values]
         </met:Values>
      </met:UpdateIntegrityReport>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The response indicates the update succeeded. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Body>
      <UpdateIntegrityReportResponse xmlns="http://metegrity.com">
         <UpdateIntegrityReportResult>true</UpdateIntegrityReportResult>
         <ErrorMsg>Operation successful.</ErrorMsg>
      </UpdateIntegrityReportResponse>
   </s:Body>
</s:Envelope>
```

IntegrityReportExistsQuery

Call without Debug Information

```
<!-- This simple call verifies that a Fixed integrity report (identified by both MemoType
     and MemoNo) exists in the database. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:met="http://metegrity.com">
   <soapenv:Header/>
   <soapenv:Body>
      <met:IntegrityReportExistsQuery>
         <met:SiteName>[Visions SiteName]</met:SiteName>
         <met:UserName>[Visions UserName]</met:UserName>
         <met:Password>[Visions Password]</met:Password>
         <met:MemoNo>1414</met:MemoNo>
         <met:MemoType>IR</met:MemoType>
      </met:IntegrityReportExistsQuery>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- Yes, it does. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Body>
      <IntegrityReportExistsQueryResponse xmlns="http://metegrity.com">
         <IntegrityReportExistsQueryResult>true</IntegrityReportExistsQueryResult>
         <existsResult>true</existsResult>
         <ErrorMsg>Operation successful.</ErrorMsg>
      </IntegrityReportExistsQueryResponse>
   </s:Body>
</s:Envelope>
```

IntegrityReportListQuery

The IntegrityReportListQuery call simply point to MemoListQuery which implicitly restricts the selection to Memo Type = ’IT’ or Memo Type = ’WR’, depending on which call is used so that you do not have to include the MemoType within the values.

Call without Debug Information

```
<!-- Here we use IntegrityReportListQuery to retrieve all the Fixed Integrity Reports for
     two pieces of equipment. -->
<soapenv:Envelope                xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:met="http://metegrity.com"
xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
   <soapenv:Header/>
   <soapenv:Body>
      <met:IntegrityReportListQuery>
         <met:SiteName>[Visions SiteName]</met:SiteName>
         <met:UserName>[Visions UserName]</met:UserName>
         <met:Password>[Visions Password]</met:Password>
         <met:FieldName1>StatHdr.EquipType</met:FieldName1>
         <met:FieldValues1>
            <arr:string>Pipe</arr:string>
            <arr:string>Exchanger</arr:string>
         </met:FieldValues1>
         <met:FieldName2>StatHdr.PlantNum</met:FieldName2>
         <met:FieldValues2>
            <arr:string>U-1300 - Polymerization</arr:string>
            <arr:string>Plant 06 - Clear Lake Gathering</arr:string>
         </met:FieldValues2>
      </met:IntegrityReportListQuery>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The Response includes the requested data in a two-dimensional OutList array, with
    OutList[0] containing the Column Header names. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Body>
      <IntegrityReportListQueryResponse xmlns="http://metegrity.com">
         <IntegrityReportListQueryResult>true</IntegrityReportListQueryResult>
         <OutList xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
          xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
            <a:ArrayOfstring>
               <a:string>Plant ID</a:string>
               <a:string>Equipment Type</a:string>
               <a:string>Equipment Number</a:string>
               <a:string>Associated Major</a:string>
               <a:string>IR Memo No</a:string>
               <a:string>Type</a:string>
               <a:string>Date Created</a:string>
               <a:string>Title</a:string>
               <a:string>Created By</a:string>
               <a:string>Date Completed</a:string>
               <a:string>Extent</a:string>
               <a:string>Stream</a:string>
               <a:string>Date Inspected</a:string>
               <a:string>Job File Number</a:string>
               <a:string>Work Order No.</a:string>
               <a:string>WE Name</a:string>
               <a:string>Report Number</a:string>
               <a:string>Reason</a:string>
               <a:string>Assigned User</a:string>
            </a:ArrayOfstring>
            <a:ArrayOfstring>
               <a:string>Plant 06 - Clear Lake Gathering</a:string>
               <a:string>Pipe</a:string>
               <a:string>CL-WHS-200</a:string>
               <a:string i:nil="true"/>
               <a:string>332</a:string>
               <a:string>IR</a:string>
               <a:string>02/10/2007</a:string>
               <a:string>Initial API 570 external Piping Inspection</a:string>
               <a:string i:nil="true"/>
               <a:string>10/06/2008</a:string>
               <a:string>Partial</a:string>
               <a:string>ExternalOff</a:string>
               <a:string>09/06/2008</a:string>
               <a:string i:nil="true"/>
               <a:string i:nil="true"/>
               <a:string>2008 Start Up</a:string>
               <a:string i:nil="true"/>
               <a:string>Routine</a:string>
               <a:string i:nil="true"/>
```

```
</a:ArrayOfstring>
<a:ArrayOfstring>
   <a:string>Plant 06 - Clear Lake Gathering</a:string>
   <a:string>Pipe</a:string>
   <a:string>CL-WHS-300</a:string>
   <a:string i:nil="true"/>
   <a:string>346</a:string>
   <a:string>IR</a:string>
   <a:string>02/10/2007</a:string>
   <a:string>Initial API 570 external Piping Inspection</a:string>
   <a:string i:nil="true"/>
   <a:string>10/06/2008</a:string>
   <a:string>Partial</a:string>
   <a:string>ExternalOff</a:string>
   <a:string>09/06/2008</a:string>
   <a:string i:nil="true"/>
   <a:string i:nil="true"/>
   <a:string>2008 Start Up</a:string>
   <a:string i:nil="true"/>
   <a:string>Routine</a:string>
   <a:string i:nil="true"/>
</a:ArrayOfstring>
<a:ArrayOfstring>
   <a:string>Plant 06 - Clear Lake Gathering</a:string>
   <a:string>Pipe</a:string>
   <a:string>CL-WHS-400</a:string>
   <a:string i:nil="true"/>
   <a:string>371</a:string>
   <a:string>IR</a:string>
   <a:string>02/10/2007</a:string>
   <a:string>API 570 external Piping Inspection</a:string>
   <a:string i:nil="true"/>
   <a:string>01/10/2014</a:string>
   <a:string>10/06/2008</a:string>
   <a:string>Partial</a:string>
   <a:string>ExternalOff</a:string>
   <a:string>09/16/2008</a:string>
   <a:string i:nil="true"/>
   <a:string i:nil="true"/>
   <a:string>2008 Start Up</a:string>
   <a:string i:nil="true"/>
   <a:string>Routine</a:string>
   <a:string i:nil="true"/>
</a:ArrayOfstring>
<a:ArrayOfstring>
   <a:string>U-1300 - Polymerization</a:string>
   <a:string>Exchanger</a:string>
   <a:string>E-1301A</a:string>
   <a:string>E-1301A</a:string>
   <a:string>604</a:string>
   <a:string>IR</a:string>
   <a:string>25/08/2011</a:string>
   <a:string>Full Visual and Internal Inspection</a:string>
   <a:string i:nil="true"/>
   <a:string>25/05/2012</a:string>
   <a:string>25/08/2012</a:string>
   <a:string>Full</a:string>
   <a:string>Internal</a:string>
   <a:string i:nil="true"/>
```

```
               <a:string>384351</a:string>
               <a:string>2012 Start Up</a:string>
               <a:string>268485</a:string>
               <a:string>Routine</a:string>
               <a:string i:nil="true"/>
            </a:ArrayOfstring>
            <a:ArrayOfstring>
               <a:string>U-1300 - Polymerization</a:string>
               <a:string>Exchanger</a:string>
               <a:string>E-1302A</a:string>
               <a:string>E-1302A</a:string>
               <a:string>605</a:string>
               <a:string>IR</a:string>
               <a:string>25/08/2011</a:string>
               <a:string> Full Visual and Internal Inspection</a:string>
               <a:string i:nil="true"/>
               <a:string>25/08/2012</a:string>
               <a:string>Full</a:string>
               <a:string>Internal</a:string>
               <a:string>30/05/2012</a:string>
               <a:string i:nil="true"/>
               <a:string>384351</a:string>
               <a:string>2012 Start Up</a:string>
               <a:string>268485</a:string>
               <a:string>Routine</a:string>
               <a:string i:nil="true"/>
            </a:ArrayOfstring>
         </OutList>
         <moreLeft>false</moreLeft>
         <ErrorMsg>Operation successful.</ErrorMsg>
      </IntegrityReportListQueryResponse>
   </s:Body>
</s:Envelope>
```

Additional Information

Visions Enterprise Help File

Lookup Data > External Comments

Lookup Data > F3 Lookup Lists

Lookup Data > Integrity Report (IR) Actions

Work Cycle > Integrity Reports > Fixed IR > Fixed IR Detail

Visions Administrator

Settings > Required Fields

Dashboard API

Overview

Several application program interfaces (API) are available to allow you to perform certain actions without using the Visions Enterprise user interface (UI). This allows third party developers to use web service calls to get or update Visions data.

The Visions APIs for dashboards allows you to get dashboard counts and datasets for both the user and main dashboards available within the Visions Client.

Access

Navigation

The web service calls are expressed as a C# interface, with the individual calls, conventions, references, and examples documented below.

```
public interface IVisAPI {
...
}
```

Conditions

1. The Dashboard API is a separate module which must be active to use its calls.
2. Dashboard items are dependent upon specific modules which must also be active to return the results:
   a. TMLs, Scheduling, and Work items are part of the Base module.
   b. Trending items are part of the Trending module.
   c. Advanced CP items are part of the Advanced CP module.
   d. IOW items are part of the IOW (Integrity Operating Windows) module.
   e. QA items are part of the Quality Assurance module.

Input

The Dashboard API provides the following calls:

◢ **QueryDashboard**: query the main dashboard, returning all the dashboard items and their respective counts based on the provided starting and ending date range

Limitation: The results are based on the plant and equipment type security access and the Equipment Index Filtered Settings configured for the username used within the call.

- **QueryDashboardList**: query the main or user dashboards, returning a list of index (log) information fields based on the provided dashboard item and the starting and ending date range
- **QueryUserDashboard**: query the user dashboard, returning all the dashboard items and their respective counts based on the queried user name

  Note: You may specify a user name that is different than the user name used to connect to the site.

Calls

```
bool QueryDashboard(string SiteName, string UserName, string Password, string Start,
    string End, string DateFmt, out string DashboardData, out string ErrorMsg);
bool QueryUserDashboard(string SiteName, string UserName, string Password, string
    QueryUser, out string DashboardData, out string ErrorMsg);
bool QueryDashboardList(string SiteName, string UserName, string Password, string
    FieldName, string Start, string End, string DateFmt, string QueryUser, string
    ColumnSet, int pageSize, int pageNum, out string[][] DashboardList, out bool moreLeft,
    out string ErrorMsg);
```

Calling Conventions

The web service is a standard SOAP web service and may be consumed by anything that can consume SOAP web services.

The following conventions are specific to the Dashboard API:

- The API web service ensures that the user account has the correct functional security permissions to perform the associated operation (i.e.: Open TML Trending).
- The URL for the web service's Basic HTTP Endpoint is:

  http://[servername]:7137/VisAPI/VisAPI_BH
- The URL to retrieve WSDL for the web service is as follows:

  http://[servername]:7137/VisAPI?wsdl

  Where [ServerName] is the name of your Visions application server (aka middle tier)

Dashboard Data Format

The QueryDashboard function returns a string value, DashboardData, which contains all the dashboard category items and names, and the count values. This data is encoded as a comma-separated-value string, which can thus easily be split into an array of strings. Each field is represented by three strings: a dashboard category string, a dashboard item label and a count value. Fields which have the same category string belong to the same category, and should be grouped together accordingly when displayed. The specific fields returned are not defined within this call.

The QueryUserDashboard function returns the user Dashboard data in the same format as the QueryDashboard function.

Note: You may use the QueryUsers function described in the main Visions API chapter to query a full list of the available users for a site.

The QueryDashboardList function returns a list of string values for each record found within the start and end date parameters, with the first array of string values provided the field labels and the remaining arrays providing each record. Null values are also returned as "i:nil="true"".

XML Field Values

The API calls accept values for arbitrary fields in a very simple XML format, illustrated below and in the examples section:

```
<met:Start>08-16-2010</met:Start>

<met:End>08-16-2014</met:End>

<met:DateFmt>MM-dd-yyyy</met:DateFmt>
```

The only significant attributes are "Start", "End" and "fmt". This allows the specification of the starting and ending date range for the dashboard counts.

All date fields require that a format string be included (in the "fmt" attribute) with their value, so that we are able to interpret the value correctly and pre-emptively avoid any issues with date formatting, time zones, locale settings and so forth. The syntax of the date format string is the .NET standard, and is described here:

http://msdn.microsoft.com/en-us/library/az4se3k1

and here:

http://msdn.microsoft.com/en-us/library/8kb3ddd4

References

Dashboard Field Name Items

The dashboard list query call allows you to specify which category and item you want query from the main dashboard to return a list of records within the specified dates.

The following list of categories and items are the values that may be used as the "FieldName" parameter (`<met:FieldName>TMLs/Expiring</met:FieldName>`):

- Alerts as of Today
  - Last Min. Read. <= Crit. Thick.
  - Last Min. Read. <= Min. Thick.
  - Expired TMLs
  - Trend Point Last Read <= Minimum
  - Equipment RLF <= Last Insp Date
- TMLs
  - RI
  - DEG
  - Expiring

  - Quarantined
- Scheduling
  - Next Full Insp
  - Components
  - External Insp
  - Next Service
- Work
  - Inspection Tasks
  - Work Requests
  - Recommendations
  - Equip NCRs
  - Other NCRs
- Trending
  - General Trending
  - TVT
- Advanced CP
  - Test Points
- IOW Violated Tags
  - Active
  - Inactive
- QA
  - Qualifications
  - Registrations
  - Certifications
  - Audits
  - Calibrations
  - Maintenance
  - Leak Tests
- IOW Alerts As Of Now
  - Violated Tags in Last 24 hrs (Active)
  - Violated Tags in Last 24 hrs (Inactive)
  - Violated Tags in Last 48 hrs (Active)
  - Violated Tags in Last 48 hrs (Inactive)
  - Violated Tags in Last 72 hrs (Active)

- Violated Tags in Last 72 hrs (Inactive)

Dashboard Items

Defines each dashboard, identifying the return result dataset structure.

Alerts as of today

Displays alerts as of today's date using the specified date format.

Limitation: The selected range of dates does not affect these counts.

Dashboard Field Name Items	Description	Resultset
Last Min Read <= Crit. Thick	Active TMLs for in-service equipment where the Last Read Min Thick is less than or equal to the Critical Thickness	TML Trending Log (Classification, Plant ID, Equipment Type, Equipment Number, TML ID, TML Type, Section ID, TML Description, Column Count, Row Count, Circuit ID, Location, Associated Major (Equipment Number), TML Description, Corr Circuit ID, Year Installed, Nominal, Corr Allow, Min Thick, Critical Thick, Stream, Active, Insulation Removal, Scaffold Required, Manlift, Ladder Required, Confined Space, Other Access, Activity Code, Min Short Loss, Min Long Loss, Avg Short Loss, Avg Long Loss, IRC, RL Factor, Restricted Interval, TML Classification, Next Insp Date Due, Next Insp Date Calculated, Last Reading Date, Last Read Min Thick, Last Read Avg Thick, Expiration Date)
Last Min Read <= Min. Thick	Active TMLs for in-service equipment where the Last Read Min Thick is less than or equal to the Minimum Thickness	
Expired TMLs	Active TMLs for in-service equipment where the Expiration Date is less than or equal to the current date	
Trend Point Last Read <= Minimum	General Trending Test Points for in-service equipment where the last reading is less than or equal to the minimum reading	General Trending Log (Plant ID, Equipment Type, Equipment Number, Associated Major Equipment Number, Trend Test Point, Description, Trend Type, Last Test Date, Next Test Due Date, Next Test Date Calculated, Interval, Minimum Reading, Maximum Reading)
Equipment RLF <= Last Insp Date	Equipment scheduling where the Remaining Life Factor (RLF) date is less than the last full inspection date	Scheduling Log (Plant ID, Equipment Type, Equipment Number, Associated Major Equipment Number, Outage Date, Next Full Insp Due, Based On, Equipment RLF Date, Last External Date, Next Ext Due Date, External Interval, IRC, Restricted Interval, In Service Date, Install Date, CA Applied, Criticality IRC, CA RC ID, CA Risk Rank, Hazard Class)

TMLs

Dashboard items that are extracted from the TML Trending.

Dashboard Field Name Items	Description	Resultset
RI	active TMLs for in-service equipment where the TML's Next Insp Date Due is between the selected dates and the Based On value is RI	TML Trending Log (Classification, Plant ID, Equipment Type, Equipment Number, TML ID, TML Type, Section ID, TML Description, Column Count, Row Count, Circuit ID, Location, Associated Major (Equipment Number),

Expiring	active TMLs for in-service equipment where the TML's Expiration Date is between the selected dates	TML Description, Corr Circuit ID, Year Installed, Nominal, Corr Allow, Min Thick, Critical Thick, Stream, Active, Insulation Removal, Scaffold Required, Manlift, Ladder Required, Confined Space, Other Access, Activity Code, Min Short Loss, Min Long Loss, Avg Short Loss, Avg Long Loss, IRC, RL Factor, Restricted Interval, TML Classification, Next Insp Date Due, Next Insp Date Calculated, Last Reading Date, Last Read Min Thick, Last Read Avg Thick, Expiration Date)
DEG	active TMLs for in-service Equipment where the TML's Next Insp Date Due is between the selected dates and the Based On value is DEG	
Quarantined	TMLs currently stored in Quarantine (datalogger survey results) awaiting review to move to TML Trending	Plant ID, Equipment Type, Equipment Number, TML ID, Test Date, Baseline, TML Type, Row Count, Column Count, Inspector, Section ID, Last Reading Date, Last Read Min Thick, Critical Thick, Error Message

Scheduling

Dashboard items that are extracted from the Scheduling, both equipment and component.

Dashboard Field Name Items	Description	Resultset
Components	components that have a Next Insp Full Due between the selected dates	Plant ID, Component Number, Component Type, Next Insp Due Date, Next Insp Date Calc, Last Insp Date
External Insp	in-service equipment that has a Next External Due Date between the selected dates	Scheduling Log (Plant ID, Equipment Type, Equipment Number, Associated Major Equipment Number, Outage Date, Next Full Insp Due, Based On, Equipment RLF Date, Last External Date, Next Ext Due Date, External Interval, IRC, Restricted Interval, In Service Date, Install Date, CA Applied, Criticality IRC, CA RC ID, CA Risk Rank, Hazard Class)
Next Full Insp	in-service, non-PSV equipment that has a Next Insp Full Due between the selected dates	
Next Service	in-service PSV equipment that has a Next Insp Full Due (Next Service Due) between the selected dates	

Notes: Non-PSV excludes equipment types derived from the supplied PSV and PSV Location types. PSV includes only equipment types derived from the supplied PSV and PSV Location types.

Work

Dashboard items that are extracted from the Inspection Task and Work Request Work Memos, the Recommendations (Integrity Report, Service Report, RBI Assessment, or Integrity Manuals), and the Non-Conformance Reports.

Dashboard Field Name Items	Description	Resultset
Inspection Tasks	inspection tasks for in-service equipment where the inspection task's Due Date is between the selected dates, and the memo Status is open or ready	Work Memo Log (Plant ID, Equipment Type, Equipment Number, Associated Major Equipment Number, Type, Memo No, Date Created, Title, Completed, Status, Problem/Failure Code, Activity Code, Extent, Stream, Interval, Job File Number, Work Order No, Priority Code, Date Completed, Reason Closed, Due Date, WE Name, WT Number, Assigned User, Integrity Critical)
Work Requests	work requests for in-service Equipment where the work request's Due Date is between the selected	

	dates, and the memo Status is open or ready	
Recommendations	recommendations applied to in-service equipment where the recommendation’s Status is open; the selected range of dates does not affect this count	Plant ID, Equipment Type, Equipment Number, Recommendation No, Title, Category
Equip NCRs	NCRs for in-service equipment where the NCR’s Due Date is between the selected dates and the NCR is not completed	NCR Log (NCR Number, NCR Type, Plant ID, Equipment Type, Equipment Number, Title, Associated Major Equipment Number, Job File Number, Work Order No, Priority Code, Completed, Status, Issued Date, Issued To, Due Date, Recommended Action, Date Completed, Assigned User, Hazard Class, MOC Number)
Other NCRs	non-equipment related NCRs where the NCR’s Due Date is between the selected dates and the NCR is not completed	

Trending

Dashboard items that are extracted from the General Trending Trend Points and Tank Volume Trending.

Dashboard Field Name Items	Description	Resultset
General Trending	general trending test points for in-service equipment where the Next Test Due Date is between the selected dates	General Trending Log (Plant ID, Equipment Type, Equipment Number, Associated Major Equipment Number, Trend Test Point, Description, Trend Type, Last Test Date, Next Test Due Date, Next Test Date Calculated, Interval, Minimum Reading, Maximum Reading)
TVT	in-service equipment for which the tank volume trending Projected Life (Calculated) falls within the selected dates	Plant ID, Equipment Number, Equipment Type, Projected Life, Remaining Volume, Last Test Date, Flow Rate

Advanced CP

Dashboard item that is extracted from the CP Testing.

Dashboard Field Name Items	Description	Resultset
Test Points	CP surveys from CP Testing for in-service equipment where the Next Due Date is between the selected dates	CP Testing Log (Plant ID, Train/Unit, Equipment Type, Equipment Number, Equipment Name, Test Point ID, Description, Test Point Type, Reading Count, Last Test Date, Interval, Next Date, Category, Location, Diameter, Length, Width, Depth)

IOW Violated Tags

Dashboard items that are extracted from the Integrity Operating Windows violated tags.

Dashboard Field Name Items	Description	Resultset
Active	Active IOW tags where the violation readings exceed the upper or lower boundary limit and the violation date is between the selected dates	IOW Tag ID, Server Name, Description, Lower Boundary Limit, Upper Boundary Limit, Monitoring
Inactive	Inactive IOW tags where the violation readings exceed the upper or lower boundary limit and the violation date is between the selected dates	

QA

Dashboard items that are extracted from the Quality Assurance that contain due dates; Contractors, Welders, Inspectors, Inspection Equipment, and Gauges.

Dashboard Field Name Items	Description	Resultset
Qualifications	welders where the qualifications have an Expiry Date within the selected dates	Welders Log (Contact Name, Welder ID No, Certificate No, Company Name, Email, Phone No, Cell No)
Registrations	contractors where the Program Registrations have an Expiry Date within the selected dates	Contractors Log (Company Name, Contact Name, Contractor Category, Controlled Manual Available, Program Status)
Certifications	inspectors or examiners where the Certifications have an Expire Date within the selected dates	Inspectors Log (Contact Name, Contact Title, Company Name, Program Status)
Audits	audits where the Inspector/Examiner or Contractor Audits have a Next Audit Date within the selected dates	Contact Name, Contact Title, Personnel Type, Company Name, Program Status
Calibrations	inspection equipment where the Calibrations have a Next Calibration Date within the selected dates	Inspection Equipment Log (Tag Number, Item, Item Type, Model No, Serial No, In Service)
Maintenance	inspection equipment where the Maintenance have a Next Inspection Date within the selected dates	
Leak Tests	gauges where the Leak Tests have a Next Test Date within the selected dates	Gauges Log (Tag Number, Radioisotope, Radioactivity, Assay Date, Equipment Location, MSDS Link)

IOW Alerts as of now

Displays integrity operating window today's date and time using your Windows short date format and time.

Limitation: The selected range of dates does not affect these counts.

Dashboard Field Name Items	Description	Results
Violated Tags in Last 24 hrs (Active)	active IOW Tag IDs where the violation readings exceed the upper or lower boundary limit within the last 24 hours	IOW Tag ID, Server Name, Description, Lower Boundary

Violated Tags in Last 24 hrs (Inactive)	active IOW Tag IDs where the violation readings exceed the upper or lower boundary limit within the last 48 hours	Limit, Upper Boundary Limit, Monitoring
Violated Tags in Last 48 hrs (Active)	active IOW Tag IDs where the violation readings exceed the upper or lower boundary limit within the last 72 hours	
Violated Tags in Last 48 hrs (Inactive)	inactive IOW Tag IDs where the violation readings exceed the upper or lower boundary limit within the last 24 hours	
Violated Tags in Last 72 hrs (Active)	inactive IOW Tag IDs where the violation readings exceed the upper or lower boundary limit within the last 48 hours	
Violated Tags in Last 72 hrs (Inactive)	inactive IOW Tag IDs where the violation readings exceed the upper or lower boundary limit within the last 72 hours	

User Dashboard Categories and Items

The dashboard list query call allows you to specify which category and item you want query from the main dashboard to return a list of records within the specified dates.

The following list of categories and items are the values that may be used as the “FieldName” parameter (`<met:FieldName>TMLs/Expiring</met:FieldName>`):

- User Dashboard
  - Fixed IRs
  - Inspection Tasks
  - NCRs
  - PSV SRs
  - Rotating IRs
  - Work Requests

User Dashboard Items

Defines each user dashboard, identifying the return result dataset structure.

User Dashboard Field Name Items	Description	Resultset
Fixed IRs	incomplete Fixed Integrity Reports where the Assigned User is set to the selected user name	Fixed IR Log (Plant ID, Equipment Type, Equipment Number Associated Major Equipment Number, IR Memo No, Type, Date Created, Title, Created By, Completed, Extent, Stream, Date Inspected, Job File Number, Work Order No, WE Name, Date Completed, Report Number, Reason, Assigned User)
Inspection Tasks	open or ready Inspection Tasks where the Assigned User Name is set to the selected user name	Work Memo Log (Plant ID, Equipment Type, Equipment Number, Associated Major Equipment Number, Type, Memo No, Date Created, Title, Completed, Status, Problem/Failure Code, Activity Code, Extent, Stream, Interval, Job File Number, Work Order No, Priority Code, Date Completed, Reason Closed, Due Date, WE Name, WT Number, Assigned User, Integrity Critical)
NCRs	incomplete Nonconformance Reports where the Assigned User Name is set to the selected user name	NCR Log (NCR Number, NCR Type, Plant ID, Equipment Type, Equipment Number, Title, Associated Major Equipment Number, Job File Number, Work Order No, Priority Code, Completed, Status, Issued Date, Issued To, Due Date,

		Recommended Action, Date Completed, Assigned User, Hazard Class, MOC Number)
PSV SRs	incomplete Service Reports where the Assigned User is set to the selected user name	Service Report Log (Plant ID, Equipment Type, Equipment Number, Service No, Date Serviced, Report Number, Title, Company, Completed, Extent, Visual External, Test Result, Assigned User, WE Name)
Rotating IRs	incomplete Rotating Integrity Reports where the Assigned User is set to the selected user name	Rotating IR Log (Plant ID, Equipment Type, Equipment Number Associated Major Equipment Number, IR Memo No, Type, Date Created, Title, Created By, Completed, Extent, Stream, IR Category, Date Inspected, Work Order No, WE Name, Date Completed, Report Number, Job File No, Reason, Assigned User)
Work Requests	open or ready Work Requests where the Assigned User Name is set to the selected user name	Work Memo Log (Plant ID, Equipment Type, Equipment Number, Associated Major Equipment Number, Type, Memo No, Date Created, Title, Completed, Status, Problem/Failure Code, Activity Code, Extent, Stream, Interval, Job File Number, Work Order No, Priority Code, Date Completed, Reason Closed, Due Date, WE Name, WT Number, Assigned User, Integrity Critical)

ColumnSet Strings

The ColumnSet strings for the Log Forms that are part of the Dashboard API.

Note: These strings may be used and modified for the columnset placeholder within the calls to limit the returned DashboardList resultset string arrays.

```
private static string
selEquipmentIndex =
  "rfSuperSubLink_EquipmentID=>SSL;rfMajorEquipment=>ME;rfSchedule=>S;rfDesMat=>D;"+
  "UnitID,PlantNum,TrainNum,kvEquipType,EquipNum,EquipName,Location,ERPNumber,"+
  "RegistrationNo,InService,Status,OutageExtent,ManufSerialNo,D.MAWP,D.TempMAWT,D.MDMT,"+
  "D.MDMTPress,D.OperatingPressure,D.OperatingTemp,S.InspRiskCode,S.RestrictedInterval,"+
  "S.NextInspFullDue,S.NextExtDueDate,"+
  "S.RLFactor,S.OutageDate,D.PaintCode,SSL.ME.EquipNum:Associated_Major,"+
  "ApplicationType,S.BasedOn,S.PointExpireDate,S.LastExtDate,"+
  "S.ExtInterval,S.InServiceDate,S.InstallDate,"+
  "S.CAApplied,S.CARiskCode",
selTMLTrendingLog =
  "rfEquipment=>E;rfSuperSubLink_EquipmentID=>SSL;rfMajorEquipment=>ME;"+
  "E.SubEquipment,E.PlantNum,E.kvEquipType,E.EquipNum,"+
  "TMLID,Grid,SectionID,PointDescription,GridColumns,GridRows,CircuitEquipmentID,"+
  "E.Location,E.SSL.ME.EquipNum:Associated_Major,PointDescription,CCID,SectionID,"+
  "YearInstalled,NomThick,CorrAllow,MinThick,CriticalThick,OnOffFlag,ActiveFlag,"+
  "InsulRemoval,ScaffReqrd,Manlift,LadderReqrd,ConfinedSpace,OtherAccess,"+
  "ActivityCode,MinShortLoss,MinLongLoss,AveShortLoss,AveLongLoss,InspRiskCode,"+
  "RLFactor,RestrictedInterval,PointClassification,NextInspDate,CalcNextInsp,"+
  "LastReadingDate,LastReadMinThick,LastReadAveThick,ExpirationDate",
selTMLTrendingLog =
  "rfEquipPoint=>P;rfEquipment=>E;rfSuperSubLink_EquipmentID=>SSL;rfMajorEquipment=>ME;"+
  "P.E.SubEquipment,P.E.PlantNum,P.E.kvEquipType,P.E.EquipNum,"+
  "TMLID,Grid,P.SectionID,P.PointDescription,GridColumns,GridRows,P.CircuitEquipmentID,"+
  "P.E.Location,P.E.SSL.ME.EquipNum:Associated_Major,PointDescription,P.CCID,"+
  "P.SectionID,P.YearInstalled,P.NomThick,P.CorrAllow,P.MinThick,P.CriticalThick,"+
  "P.OnOffFlag,P.ActiveFlag,P.InsulRemoval,P.ScaffReqrd,P.Manlift,P.LadderReqrd,"+
  "P.ConfinedSpace,P.OtherAccess,P.ActivityCode,P.MinShortLoss,P.MinLongLoss,"+
  "P.AveShortLoss,P.AveLongLoss,P.InspRiskCode,"+
  "P.RLFactor,P.RestrictedInterval,P.PointClassification,P.NextInspDate,P.CalcNextInsp,"+
  "P.LastReadingDate,P.LastReadMinThick,P.LastReadAveThick,P.ExpirationDate",
selSchedulingIndex =
  "rfEquipment=>E;rfSuperSubLink_EquipmentID=>SSL;rfMajorEquipment=>ME;"+
  "E.PlantNum,E.kvEquipType,E.EquipNum,E.SSL.ME.EquipNum:Associated_Major,"+
  "OutageDate,NextInspFullDue,BasedOn,PointExpireDate,LastExtDate,NextExtDueDate,"+
  "ExtInterval,InspRiskCode,RestrictedInterval,InServiceDate,InstallDate,"+
  "CAApplied,CARiskCode",
selComponentIndex =
  "rfCompSchedule=>S;PlantNum,kvCompType,ComponentNum,ComponentDescription,"+
  "ComponentFunction,ERPNumber,InUse,Location,OutageExtent,YearInstalled,"+
  "FlowDiagram,PIDNumber,SerialNumber,ScaffoldReqd,ManliftReqd,LadderReqd,"+
  "S.InspRiskCode,S.NextInspFullDue,S.NextInspFullCalc",
selWorkMemoIndex =
  "rfEquipment=>E;rfSuperSubLink_EquipmentID=>SSL;rfMajorEquipment=>ME;"+
  "E.PlantNum,E.kvEquipType,E.EquipNum,E.SSL.ME.EquipNum:Associated_Major,"+
  "MemoType,MemoNo,DateCreated,MemoTitle,Completed,CloseFlag,ProblemCode,"+
  "ActivityCode,FullPartFlag,OnOffFlag,RInterval,JobFileNo,WorkOrderNo,"+
  "PriorityCode,DateCompleted,CloseReason,DueDate,rfWorkEvent.WENumber:WE_Name,"+
  "rfItemNumber.WPNumber:WT_Number,AssignedUser",
selRecomIndex =
  "rfEquipment=>E;rfRecom=>R;"+
  "E.PlantNum,E.kvEquipType,E.EquipNum,R.RecommNum,R.MemoTitle,R.Category",
selEquipNCRIndex =
  "rfEquipment=>E;rfSuperSubLink_EquipmentID=>SSL;rfMajorEquipment=>ME;"+
  "NCRNumber,E.EquipNum,E.PlantNum,E.kvEquipType,E.SSL.ME.EquipNum:Associated_Major,"+
  "WorkOrderNo,JobFileNum,MemoTitle,PriorityCode,Completed,CloseFlag,IssuedDate,"+
  "IssuedTo,DispDate,NCRType,RecAction,QcComplDate,AssignedUser,E.HazardClass",
selOtherNCRIndex =
  "NCRNumber,"+
  "WorkOrderNo,JobFileNum,MemoTitle,PriorityCode,Completed,CloseFlag,IssuedDate,"+
  "IssuedTo,DispDate,NCRType,RecAction,QcComplDate,AssignedUser",
selCPSurveys =
  "rfEquipment=>E;E.PlantNum,E.EquipNum,E.kvEquipType,NextCPDue,NextCPCalc,"+
  "LastCPDate,CPInterval",
selTrendingLog =
  "rfEquipment=>E;rfSuperSubLink_EquipmentID=>SSL;rfMajorEquipment=>ME;"+
  "E.PlantNum,E.kvEquipType,E.EquipNum,E.SSL.ME.EquipNum:Associated_Major,"+
  "kvmTrendTestPoint,Description,kvTrendType,LastDate,NextDate,NextCalc,"+
  "RInterval,MinRead,MaxRead",
selTankVolTrend =
  "rfEquipment=>E;E.PlantNum,E.EquipNum,E.kvEquipType,ProjectedLife,LastTestDate,"+
  "FlowRate,RemainVol",
selAdvancedCP =
  "rfEquipment=>E;E.PlantNum,E.TrainNum,E.EquipNum,E.kvEquipType,TPID,TPType,"+
  "E.EquipName,Description,ReadCount,LastDate,RInterval,NextDate,Category,"+
  "Location,Diameter,Length,Width,Depth",
selCBIMTags =
  "rfCBIM_Tags=>T;T.CBIMTagID,T.kvPIServerName,T.Description,T.LowBoundLimit,"+
  "T.UpBoundLimit,T.ActiveFlag:Monitoring",
selQAQualifications =
  "ContactName,rfQAWelders.WelderIDNo,rfQAWelders.CertificateNo,CompanyName,Email,"+
  "PhoneNo,CellNo",
selQARegistrations =
  "CompanyName,ContactName,rfQAContractors.ContractorCategory:Contractor_Category,"+
  "rfQAContractors.ControlledManualAvail,ProgramStatus",
selQACertifications =
  "ContactName,ContactTitle,CompanyName,ProgramStatus",
selQAAudits =
  "ContactName,ContactTitle,PersonnelType,CompanyName,ProgramStatus",
selQACalibrations =
  "TagNumber,mItem,rfItemType.ItemType,ModelNo,SerialNo,Status",
selQALeakTests =
  "rfGauge=>G;G.TagNumber,G.Radioisotope,G.Radioactivity,G.AssayDate,G.EquipNumber,G.MSDS
  Link",
selQAMaintenance =
  "TagNumber,mItem,rfItemType.ItemType,ModelNo,SerialNo,Status",
selFixedIR =
  "rfEquipment=>E;rfSuperSubLink_EquipmentID=>SSL;rfMajorEquipment=>ME;"+
  "E.PlantNum,E.kvEquipType,E.EquipNum,E.SSL.ME.EquipNum:Associated_Major,"+
  "MemoNo,MemoType,DateCreated,MemoTitle,CreatedBy,DateCompleted,FullPartFlag,"+
  "OnOffFlag,InspDate,JobFileNo,WorkOrderNo,rfWorkEvent.WENumber:WE_Name,"+
  "DateCompleted,ReportNumber,Reason,AssignedUser",
selRotatingIR =
  "rfEquipment=>E;rfSuperSubLink_EquipmentID=>SSL;rfMajorEquipment=>ME;"+
  "E.PlantNum,E.kvEquipType,E.EquipNum,E.SSL.ME.EquipNum:Associated_Major,"+
  "MemoNo,MemoType,DateCreated,MemoTitle,CreatedBy,DateCompleted,FullPartFlag,"+
  "OnOffFlag,InspDate,JobFileNo,WorkOrderNo,rfWorkEvent.WENumber:WE_Name,"+
  "DateCompleted,ReportNumber,Reason,AssignedUser",
selPSVSR =
  "rfEquipment=>E;E.PlantNum,E.kvEquipType,E.EquipNum,ServiceNum,DateServiced,"+
  "ReportNum,MemoTitle,Company,Completed,FullPartFlag,IntExtFlag,"+
  "rfWorkEvent.WENumber:WE_Name,TestResult,AssignedUser",
selQueryCPTPs =
  "rfEquipment=>E;E.PlantNum,E.kvEquipType,E.EquipNum,TPID,Category,Depth,Description,"+
  "Diameter,DrawingNo,LastDate,LastUpdate,Length,Location,Manufacturer,Material,"+
  "ModelNo,NextCalc,NextDate,ReadCount,RInterval,RIntervalUnit,TPType,UpdatedBy,"+
  "Width,lstCP_TPDate.kvTestDate,lstCP_TPDate.Baseline,lstCP_TPDate.FunctionalLocation,"+
  "lstCP_TPDate.OnOffFlag,lstCP_TPDate.OrigDueDate,lstCP_TPDate.TestedBy,"+
  "lstCP_TPDate.lstCP_TPReadDet.CPState,lstCP_TPDate.lstCP_TPReadDet.Distance,#0,#1,"+
  "lstCP_TPNotes.TestDate:Most_Recent_Notes_Test_Date,lstCP_TPNotes.NotesCategory,"+
  "lstCP_TPNotes.Notes",
selQueryGTTPs =
  "rfEquipment=>E;E.PlantNum,E.kvEquipType,E.EquipNum,kvmTrendTestPoint,"+
  "Description,kvTrendType,LastDate,LastUpdate,MaxRead,MinRead,NextCalc,"+
  "NextDate,ReadingUnit,RInterval,RIntervalUnit,lstTrendDates.kvTestDate,"+
  "lstTrendDates.Baseline,lstTrendDates.Comments,lstTrendDates.FunctionalLocation,"+
  "lstTrendDates.OnOffFlag,lstTrendDates.TestedBy,#0"
;
```

Call Examples

QueryDashboard

Call

```
<!-- This query returns all the Dashboard query counts for the database given Start and
    End Dates of August 16th, 2010 and August 16th, 2014. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:QueryDashboard>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:Start>08-16-2010</met:Start>
    <met:End>08-16-2014</met:End>
    <met:DateFmt>MM-dd-yyyy</met:DateFmt>
   </met:QueryDashboard>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The response is a string in comma-separated value format that includes all the
    headers, subheaders and query counts as discussed under 'Dashboard Data Format'. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <QueryDashboardResponse xmlns="http://metegrity.com">
    <QueryDashboardResult>true</QueryDashboardResult>
       <DashboardData>TMLs,RI,7,TMLs,DEG,3,TMLs,Expiring,2,TMLs,Quarantined,0,Scheduling,
       Next Full Insp,2,Scheduling,Components,0,Scheduling,External
       Insp,8,Scheduling,Next Service,0,Work,Inspection Tasks,3,Work,Work
       Requests,0,Work,Recommendations,6,Work,Equip NCRs,0,Work,Other
       NCRs,1,Trending,General Trending,0,Trending,TVT,0,Advanced CP,Test
       Points,0,IOW,Violated Tags (Active),0,IOW,Violated Tags
       (Inactive),1,QA,Qualifications,0,QA,Registrations,0,QA,Certifications,0,QA,Audits,
       3,QA,Calibrations,0,QA,Maintenance,0,QA,Leak Tests,0,Alerts as of Today,Last Min.
       Read. &lt;= Crit. Thick.,1,Alerts as of Today,Last Min. Read. &lt;= Min.
       Thick.,0,Alerts as of Today,Expired TMLs,2,Alerts as of Today,Trend Point Last
       Read &lt;= Minimum,0,IOW Alerts As Of Now,Violated Tags in Last 24 hrs
       (Active),0,IOW Alerts As Of Now,Violated Tags in Last 24 hrs (Inactive),0,IOW
       Alerts As Of Now,Violated Tags in Last 48 hrs (Active),0,IOW Alerts As Of
       Now,Violated Tags in Last 48 hrs (Inactive),0,IOW Alerts As Of Now,Violated Tags
       in Last 72 hrs (Active),0,IOW Alerts As Of Now,Violated Tags in Last 72 hrs
       (Inactive),0,List Query,Equipment,9896,List Query,Memos,1647,List Query,Work
       Requests,1048,List Query,Inspection Tasks,599</DashboardData>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </QueryDashboardResponse>
 </s:Body>
</s:Envelope>
```

QueryUserDashboard

Call

```
<!-- This call returns the User Dashboard values for the user 'BRIAN'. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:QueryUserDashboard>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:QueryUser>BRIAN</met:QueryUser>
   </met:QueryUserDashboard>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The same format is used as with QueryDashboard, above. -->
```

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <QueryUserDashboardResponse xmlns="http://metegrity.com">
    <QueryUserDashboardResult>true</QueryUserDashboardResult>
      <DashboardData>User Dashboard,Fixed IRs,1,User Dashboard,Rotating IRs,0,User
      Dashboard,PSV SRs,0,User Dashboard,Inspection Tasks,0,User Dashboard,Work
      Requests,0,User Dashboard,NCRs,0</DashboardData>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </QueryUserDashboardResponse>
 </s:Body>
</s:Envelope>
```

QueryDashboardList

Call using TMLs/Expiring

```
<!-- This query returns a list of index information fields about all the expiring TMLs in
    the date range of August 16th, 2000 to August 21st, 2014. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:QueryDashboardList>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:FieldName>TMLs/Expiring</met:FieldName>
    <met:Start>08-16-2000</met:Start>
    <met:End>08-21-2014</met:End>
    <met:DateFmt>MM-dd-yyyy</met:DateFmt>
    <met:QueryUser/>
    <met:pageSize>-1</met:pageSize>
    <met:pageNum>-1</met:pageNum>
   </met:QueryDashboardList>
 </soapenv:Body>
</soapenv:Envelope>
```

Response from TMLs/Expiring

```
<!-- As with other queries, the first row of the array in the response gives the column
    names, while data records are gathered after. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
   xmlns:i="http://www.w3.org/2001/XMLSchema-instance"> <s:Body>
   <QueryDashboardListResponse xmlns="http://metegrity.com">
    <QueryDashboardListResult>true</QueryDashboardListResult>
    <DashboardList
      <a:ArrayOfstring>
       <a:string>Classification</a:string>
       <a:string>Plant ID</a:string>
       <a:string>Equipment Type</a:string>
       <a:string>Equipment Number</a:string>
       <a:string>TML ID</a:string>
       <a:string>TML Type</a:string>
       <a:string>Section ID</a:string>
       <a:string>TML Description</a:string>
       <a:string>Column Count</a:string>
       <a:string>Row Count</a:string>
       <a:string>Circuit ID</a:string>
```

```
        <a:string>Location</a:string>
        <a:string>Associated Major</a:string>
        <a:string>TML Description</a:string>
        <a:string>Corr Circuit ID</a:string>
        <a:string>Section ID</a:string>
        <a:string>Year Installed</a:string>
        <a:string>Nominal</a:string>
        <a:string>Corr Allow</a:string>
        <a:string>Min Thick</a:string>
        <a:string>Critical Thick</a:string>
        <a:string>Stream</a:string>
        <a:string>Active</a:string>
        <a:string>Insulation Removal</a:string>
        <a:string>Scaffold Required</a:string>
        <a:string>Manlift</a:string>
        <a:string>Ladder Required</a:string>
        <a:string>Confined Space</a:string>
        <a:string>Other Access</a:string>
        <a:string>Activity Code</a:string>
        <a:string>Min Short Loss</a:string>
        <a:string>Min Long Loss</a:string>
        <a:string>Avg Short Loss</a:string>
        <a:string>Avg Long Loss</a:string>
        <a:string>IRC</a:string>
        <a:string>RL Factor</a:string>
        <a:string>Restricted Interval</a:string>
        <a:string>TML Classification</a:string>
        <a:string>Next Insp Date Due</a:string>
        <a:string>Next Insp Date Calculated</a:string>
        <a:string>Last Reading Date</a:string>
        <a:string>Last Read Min Thick</a:string>
        <a:string>Last Read Avg Thick</a:string>
        <a:string>Expiration Date</a:string>
       </a:ArrayOfstring>
       <a:ArrayOfstring>
        <a:string>Equipment</a:string>
        <a:string>U-3000 - Demonstration DB</a:string>
        <a:string>Vessel</a:string>
        <a:string>D08-V-102</a:string>
        <a:string>D08-V-102-050</a:string>
        <a:string>Scan</a:string>
        <a:string>N2</a:string>
        <a:string>2" Liquid O/L at Bottom Head</a:string>
        <a:string>2</a:string>
        <a:string>1</a:string>
        <a:string i:nil="true"/>
        <a:string>South of the Compressor Building</a:string>
        <a:string i:nil="true"/>
        <a:string>2" Liquid O/L at Bottom Head</a:string>
        <a:string i:nil="true"/>
        <a:string>N2</a:string>
        <a:string>2000</a:string>
        <a:string>0.218</a:string>
        <a:string>0.125</a:string>
        <a:string>0.093</a:string>
        <a:string>0.1023</a:string>
        <a:string>ExternalOn</a:string>
        <a:string>True</a:string>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string>UTS</a:string>
        <a:string>0</a:string>
        <a:string>0.0081</a:string>
        <a:string>0.0733</a:string>
        <a:string>0.0081</a:string>
        <a:string>3</a:string>
        <a:string>0.5</a:string>
        <a:string>12</a:string>
        <a:string>NOZ</a:string>
        <a:string>12/02/2013</a:string>
        <a:string>12/02/2013</a:string>
        <a:string>13/02/2013</a:string>
        <a:string>0.1</a:string>
        <a:string>0.1</a:string>
        <a:string>12/02/2013</a:string>
       </a:ArrayOfstring>
    </DashboardList>
    <moreLeft>false</moreLeft>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </QueryDashboardListResponse>
 </s:Body>
</s:Envelope>
```

Call using UserDashboard/Fixed IRs

```
<!-- This example shows how QueryDashboardList can be used to retrieve values for the
    User Dashboard instead of the main one. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:QueryDashboardList>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:FieldName>UserDashboard/FixedIRs</met:FieldName>
    <met:Start>08-16-2010</met:Start>
    <met:End>08-21-2016</met:End>
    <met:DateFmt>MM-dd-yyyy</met:DateFmt>
    <met:QueryUser>JULIAN</met:QueryUser>
    <met:pageSize>-1</met:pageSize>
    <met:pageNum>-1</met:pageNum>
   </met:QueryDashboardList>
 </soapenv:Body>
</soapenv:Envelope>
```

Response from UserDashboard/Fixed IRs

```
<!-- The results of a User Dashboard list query are very similar in format to those
    associated with the main Dashboard. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <QueryDashboardListResponse xmlns="http://metegrity.com">
    <QueryDashboardListResult>true</QueryDashboardListResult>
    <DashboardList xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
       xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
       <a:ArrayOfstring>
        <a:string>Plant ID</a:string>
        <a:string>Equipment Type</a:string>
        <a:string>Equipment Number</a:string>
        <a:string>Associated Major</a:string>
        <a:string>IR Memo No</a:string>
        <a:string>Type</a:string>
        <a:string>Date Created</a:string>
        <a:string>Title</a:string>
        <a:string>Created By</a:string>
        <a:string>Date Completed</a:string>
        <a:string>Extent</a:string>
        <a:string>Stream</a:string>
        <a:string>Date Inspected</a:string>
        <a:string>Job File Number</a:string>
        <a:string>Work Order No.</a:string>
        <a:string>WE Name</a:string>
        <a:string>Date Completed</a:string>
        <a:string>Report Number</a:string>
        <a:string>Reason</a:string>
        <a:string>Assigned User</a:string>
       </a:ArrayOfstring>
       <a:ArrayOfstring>
        <a:string>Plant 31 - Power Generation</a:string>
        <a:string>Boiler</a:string>
        <a:string>31F-03</a:string>
        <a:string i:nil="true"/>
        <a:string>1400</a:string>
        <a:string>IR</a:string>
        <a:string>24/08/2015</a:string>
        <a:string>Nox Ports Inspection (Tubes, casing &amp; ducts)</a:string>
        <a:string>ARTUR</a:string>
        <a:string i:nil="true"/>
        <a:string>Partial</a:string>
        <a:string>Internal</a:string>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string>2014 Spring Turnaround</a:string>
        <a:string i:nil="true"/>
        <a:string>77</a:string>
        <a:string>Re-certification</a:string>
        <a:string>JULIAN</a:string>
       </a:ArrayOfstring>
    </DashboardList>
    <moreLeft>false</moreLeft>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </QueryDashboardListResponse>
 </s:Body>
</s:Envelope>
```

Additional Information

Visions Enterprise Help File

Dashboards > Dashboard

Dashboards > User Dashboard

TML Trending API

Overview

Several application programming interfaces (API) are available to allow you to perform certain actions without using the Visions Enterprise user interface (UI). This allows third party developers to use web service calls to get or update Visions data.

The Visions API for TML trending allows you to manage thickness monitoring locations; such as creating, updating, or querying TML identifiers, readings, and notes data.

Access

Navigation

The web service calls are expressed as a C# interface, with the individual calls, conventions, references, and examples documented below.

```
public interface IVisAPI {
...
}
```

Conditions

1. The TML API is a separate module which must be active to use its calls.
2. To set the RC ID and use its associated risk code and intervals, the RBI module must be active. Otherwise to set the Inspection Risk Code and Interval, the Inspection Code Lookup Data is used and must exist prior to the TML Creation or Update.

   Note: If an RC ID value is populated in the API call when the RBI module is active, the value will still be stored but will never be used or visible within Visions.

3. Some equipment types may not own TMLs, this includes any types derived from the following supplied types: Circuit, Major, PSV, PSV Location, Cooling Tower, RE Motor, CP System, CP Custom, and CP Rectifier.
4. Security to perform calls are dependent upon the security access for the username passed into a call. For example: if you do not have Delete TML privileges, you will not be able to call the DeleteTML call successfully. Security functions for TML Trending are:
   a. **Open/Edit TML Trending**: controls access to the TML Trending, all calls
   b. **Add or Clone TML**: controls TML creation or cloning (equivalent to copy or duplicate within the Visions Client)
   c. **Change TML Thicknesses and Codes**: required to add multiple readings or access the Edit TML Data information (section, thicknesses, risk and interval information)
   d. **Delete TML**: controls deletion of a TML

e. **Edit TML Config**: controls editing of the TML Type from Point to Scan and vice versa, and the number of rows and/or columns for the TML

Limitation: Grid TML Types cannot be changed to Point or Scan and, vice versa, Point or Scan cannot be changed to Grid.

Input

The TML API provides the following calls:

◢ **CreateTML**: create the identified thickness monitoring location(s); simulates the 'New' action on the TML Trending Log or the Create TMLs workbook mapping in the Excel Wizard

Limitation: Readings cannot be populated as part of CreateTML, you will have to use both the CreateTML and AddTMLReadings to create a TML with its associated readings.

Security: Permission is dependent upon the 'Add or Clone TML' function.

◢ **CloneTML**: creates a new TML by copying an existing TML; simulates the 'Clone' and 'Duplicate' actions on the TML Trending Log

Limitation: Readings cannot be populated as part of CloneTML, you will have to use both the CloneTML and AddTMLReadings to create a TML with its associated readings.

Security: Permission is dependent upon the 'Add or Clone TML' function.

◢ **CloneTMLWithDates**: creates a new TML and sets the next inspection due date by copying an existing TML; simulates the 'Clone' and 'Duplicate' actions on the TML Trending Log

Limitation: Readings cannot be populated as part of CloneTMLWithDates, you will have to use both the CloneTMLWithDates and AddTMLReadings to create a TML with its associated readings.

Security: Permission is dependent upon the 'Add or Clone TML' function.

◢ **UpdateTML**: update an existing thickness monitoring location, including activation and deactivation; excludes changes to readings and owning equipment

Security: For the thickness and risk code fields, permission is dependent upon the 'Change TML Thicknesses and Codes' function.

Security: Permission to modify the TML Type and/or reading count is dependent upon the 'Change TML Type' function.

Note: Depending on what is changed within a TML, the scheduling information for the TML and possibly its owning equipment may be impacted.

◢ **DeleteTML**: deletes the identified thickness monitoring location(s); simulates the 'Delete' action on the TML Trending Log

Security: Permission is dependent upon the 'Delete TML' function.

Note: Depending on whether active TML(s) are deleted, the scheduling information for the TML's owning equipment may be impacted.

◢ **QueryTML**: query the thickness monitoring locations, returning the value(s) for the attribute(s) specified within the call

- **AddTMLReadings**: add new readings to the thickness monitoring location(s); simulates the 'Add Readings' on the TML Trending Log or TML Trending detail or the TML Readings workbook mapping in the Excel Wizard

  Note: Similar to the Excel Wizard's TML Readings Import function, by default any new TML Readings are stored in the Visions Quarantine so that an authorized user must review prior to reading being added into the TML Trending.

- **EditTMLReadings**: edit existing readings for the thickness monitoring location(s); simulates the 'Edit Readings' on the TML Trending detail

  Note: Depending on whether readings for active TML(s) are edited, the scheduling information for the TML and possibly its owning equipment may be impacted.

- **DeleteTMLReadings**: delete readings from thickness monitoring location(s); simulates the 'Delete Test Date' on the TML Trending detail

  Note: Depending on whether readings for active TML(s) are edited, the scheduling information for the TML and possibly its owning equipment may be impacted.

- **AddTMLNotes**: add notes to the thickness monitoring location(s) for a specified test date

Calls

```
bool QueryTML(string SiteName, string UserName, string Password, string EquipNum, string
    PlantNum, string EquipType, string TMLID, string FieldName1, string[] FieldValues1,
    string FieldName2, string[] FieldValues2, string FieldName3, string[] FieldValues3,
    string FieldName4, string[] FieldValues4, string DateFmt, string ColumnSet, int
    pageSize, int pageNum, out string[][] OutList, out bool moreLeft, out string ErrorMsg,
    string MapID);

bool CreateTML(string SiteName, string UserName, string Password, string EquipNum, string
    PlantNum, string EquipType, string TMLID, string TMLDesc, string TMLType, int RowCount,
    int ColumnCount, string Values, out string ErrorMsg, string MapID);

bool CloneTML(string SiteName, string UserName, string Password, string SrcEquipNum, string
    SrcPlantNum, string SrcEquipType, string SrcTMLID, string[] DestEquipNum, string[]
    DestPlantNum, string[] DestEquipType, string[] DestTMLID, out string ErrorMsg);

bool CloneTMLWithDates(string SiteName, string UserName, string Password, string
    SrcEquipNum, string SrcPlantNum, string rcEquipType, string SrcTMLID, string[]
    DestEquipNum, string[] DestPlantNum, string[] DestEquipType, string[] DestTMLID,
    string dateFmt, string[] dueDates, out string ErrorMsg);

bool UpdateTML(string SiteName, string UserName, string Password, string EquipNum, string
    PlantNum, string EquipType, string TMLID, string Values, out string ErrorMsg, string
    MapID);

bool DeleteTML(string SiteName, string UserName, string Password, string EquipNum, string
    PlantNum, string EquipType, string TMLID, string Values, out string ErrorMsg, string
    MapID);

bool AddTMLReadings(string SiteName, string UserName, string Password, string EquipNum,
    string PlantNum, string EquipType, string TMLID, string Values, string ReadingsCSV,
    string TestDate, string dateFmt, out string ErrorMsg, string MapID);
```

```
bool EditTMLReadings(string SiteName, string UserName, string Password, string EquipNum,
    string PlantNum, string EquipType, string TMLID, string Values, string ReadingsCSV,
    string TestDate, string dateFmt, out string ErrorMsg, string MapID);

bool DeleteTMLReadings(string SiteName, string UserName, string Password, string EquipNum,
    string PlantNum, string EquipType, string TMLID, string TestDate, string dateFmt, string
    Values, out string ErrorMsg, string MapID);

bool AddTMLNotes(string SiteName, string UserName, string Password, string EquipNum, string
    PlantNum, string EquipType, string TMLID, string TestDate, string dateFmt, string
    Category, string Notes, string Values, out string ErrorMsg, string MapID);
```

Calling Conventions

The web service is a standard SOAP web service and may be consumed by anything that can consume SOAP web services.

The following conventions are specific to the TML API:

- API calls which reference a piece of equipment do so by its three (3) logical keys – PlantNum, EquipType, and EquipNum – not our internal Visions account identifier.
- API calls to interact with existing TMLs must reference them by the logical key of TMLID and asset identification (Plant, Equipment Type, and Equipment Number per above), which uniquely identifies a thickness monitoring location for a specific piece of equipment.
- When a new TML is created, it requires an equipment reference by logical keys and the thickness monitoring location identifier (TMLID).
- Arbitrary field values may be specified using the Values parameter, passing one or more field values as XML in a manner detailed below.
- The API web service ensures that the user account has the appropriate security permissions to perform the associated operation, (i.e., Open TML Trending, Add or Clone TML, Change TML Thicknesses and Codes, Recalculate TMLs, and Delete TML).

Behavior Notes

- The API web service will correct the casing of any values that already exist in the Visions database (i.e., TMLID, EquipNum, PlantNum, etc.) unless doing so would introduce ambiguity (i.e., conflicting records differing only in case exist), in which case it reads the values case-sensitively.
- The record creation API calls create a record in the associated parent table (EquipPoint), as well as zero or more child records in child tables (i.e, Readin, ReadGrid, etc.) needed to store values specified in the Values XML string.

List Parameters and Data Format

FieldValues: allow multiple list queries; each of which may take up to four fields in the form of a field name and list of permitted values

Note: All records matching any of the given values in each field are returned.

The list queries return two-dimensional string arrays which duplicate the grid indexes shown in the Visions Client. Row zero of the returned array has the column headers, and each row thereafter is a data row. By default, the returned columns directly mirror the relevant log in the Visions client, but they may be changed or limited by providing a ColumnSet.

ColumnSet: allows you to format the response dataset using the foreign references and field namesto join related tables within a List Query call in an API, as well as assign an alias for your foreign reference(s); for example, to return only the logical keys for equipment, specify this ColumnSet value: “PlantNum,EquipNum,EquipType”

Reference the EquipmentListQuery call example for more information.

Data Format

ReadingsCSV: TML readings are passed into and out of the API as a concatenated string of decimal values delimited by semicolons – this is essentially CSV, but a semicolon is used instead of a comma because some locales use the comma in numbers to indicate a decimal place. This string must contain exactly (TML RowCount x TML ColumnCount) numbers, with the numbers in column-first order.

Numbers in a slot may be omitted, in which case whatever is currently in that position in Visions’ database will not change. $NULL or {null} can also be specified in place of a number, in which case that reading position will be blanked in the database.

Point TMLs are treated as 1xN grids, while Scan TMLs are treated as 1x2 grids.

Query Parameters: When querying TMLs, all TMLs matching the specified criteria will be returned, along with the most recent set of readings for each returned TML in ReadingsCSV format.

Identifiers for a single TML may be explicitly specified to QueryTML using the EquipNum, EquipType, PlantNum and TMLID API parameters, or may be generically specified using FieldName1, FieldValues1 and so forth. Examples below document both usages.

XML Field Values

The API calls accept values for arbitrary fields in a very simple XML format, illustrated below and in the examples section:

```
<Values>
<Value field='PointDescription' >V-01-001 Inlet Nozzle</Value>
<Value field='POintClassification' >NZ</Value>
<Value field='NextInspDate'
 fmt='MM-dd-yyyy'>08-16-2012</Value>
</Values>
```

This could be sent to create a new TML with the specified Description, Point Classification, and Next Inspection Due Date.

The only significant tag is “Value”, and the only significant attributes are “field” and “fmt”. The field attribute is specified in the Tables and Fields group of References section below. This allows the specification of field values in child tables when a TML is created or updated through the TML API.

All date fields require that a format string be included (in the “fmt” attribute) with their value, so that we are able to interpret the value correctly and pre-emptively avoid any issues with date formatting, time zones, locale settings and so forth. The syntax of the date format string is the .NET standard, and is described here:

http://msdn.microsoft.com/en-us/library/az4se3k1

and here:

http://msdn.microsoft.com/en-us/library/8kb3ddd4

References

Data Model Overview

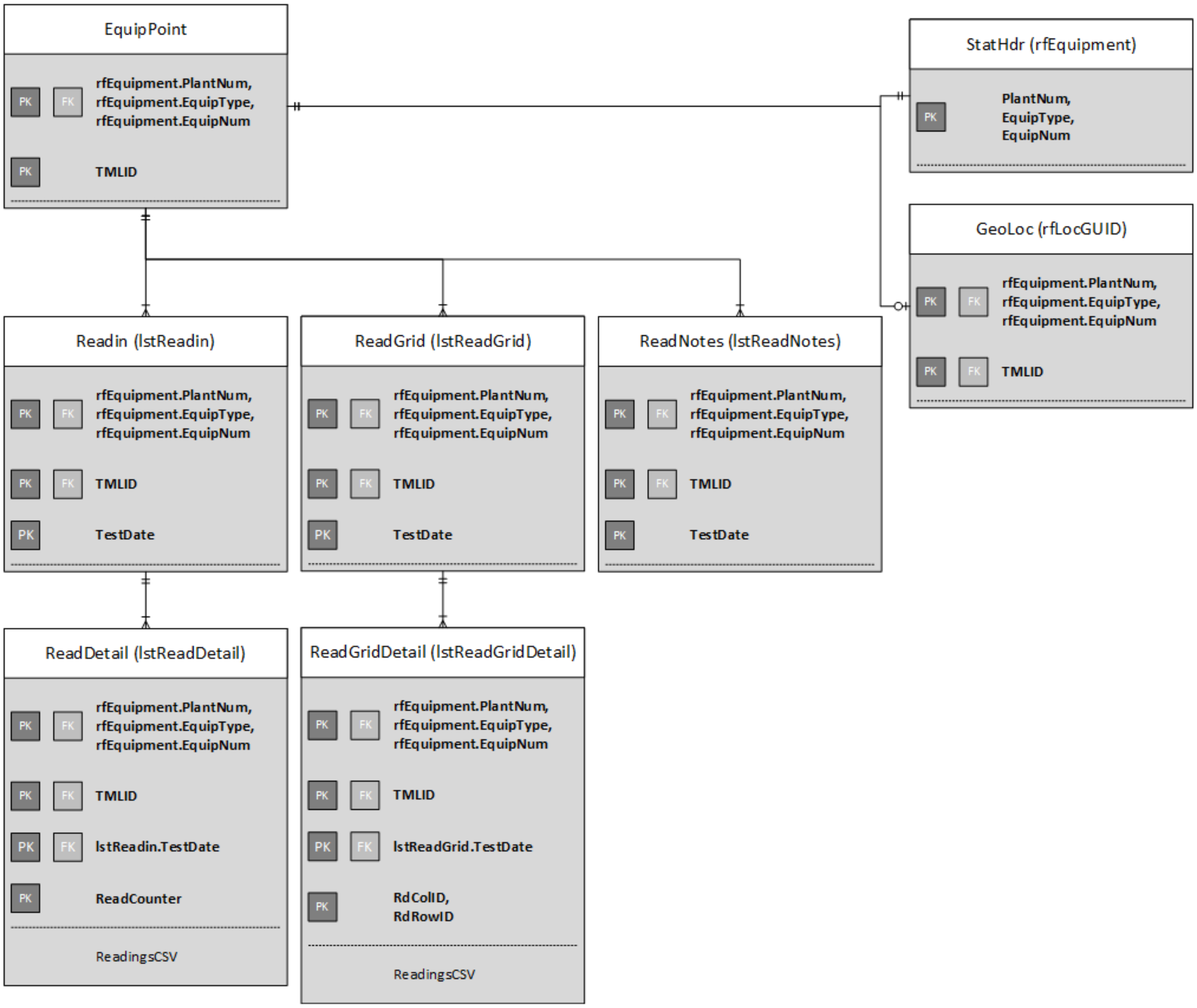

EquipPoint
PK FK rfEquipment.PlantNum, rfEquipment.EquipType, rfEquipment.EquipNum
PK TMLID
StatHdr (rfEquipment)
PK PlantNum, EquipType, EquipNum
GeoLoc (rfLocGUID)
PK FK rfEquipment.PlantNum, rfEquipment.EquipType, rfEquipment.EquipNum
PK FK TMLID
Readin (lstReadin)
PK FK rfEquipment.PlantNum, rfEquipment.EquipType, rfEquipment.EquipNum
PK FK TMLID
PK TestDate
ReadGrid (lstReadGrid)
PK FK rfEquipment.PlantNum, rfEquipment.EquipType, rfEquipment.EquipNum
PK FK TMLID
PK TestDate
ReadNotes (lstReadNotes)
PK FK rfEquipment.PlantNum, rfEquipment.EquipType, rfEquipment.EquipNum
PK FK TMLID
PK TestDate
ReadDetail (lstReadDetail)
PK FK rfEquipment.PlantNum, rfEquipment.EquipType, rfEquipment.EquipNum
PK FK TMLID
PK FK lstReadin.TestDate
PK ReadCounter
ReadingsCSV
ReadGridDetail (lstReadGridDetail)
PK FK rfEquipment.PlantNum, rfEquipment.EquipType, rfEquipment.EquipNum
PK FK TMLID
PK FK lstReadGrid.TestDate
PK RdColID, RdRowID
ReadingsCSV

Table and Field Values

The table and field structures and definitions provide the ‘Values’ string within a call. Fields requiring the data to be entered based on a site configured unit of measure are identified by “Unit-Based Field”.

Limitation: Only the tables and fields listed below are supported for the TML API.

StatHdr (rfEquipment)

Equipment Identification: identify the owning equipment within the TML calls; such as CreateTML, UpdateTML, and DeleteTML.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**EquipNum**	("Equipment Number", dtString, Required Field, Read-Only Field, Maximum Length 35)
		Displays the asset or unique identification for the equipment.
string	**EquipType**	("Equipment Type", dtString, Required Field, Read-Only Field, Maximum Length 35, FK (EquipTypeList.EquipType))
		Displays the type of equipment.
string	**PlantNum**	("Plant ID", dtString, Required Field, Read-Only Field, Maximum Length 35, FK (PlantInfo.PlantNum))
		Displays the plant where the equipment is located.

Foreign References		
Table Name	**Call Value**	**Definition**
EquipPoint	rfEquipPoint	Master table for TMLs (Thickness Monitoring Locations)

EquipPoint

TML Trending: the main TML entity that stores details for each thickness monitoring location.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**TMLID**	("TML ID", dtString, Required Field, Maximum Length 15)
		Enter the thickness monitoring location identification that uniquely identifies each monitoring location for the equipment. **Limitation**: TMLID must be unique within each owning equipment.
bool	ActiveFlag	("Active", dtInteger, Required Field)
		Indicate whether the TML is currently active. Inactive TMLs will have no calculations performed and play no part in scheduling for the equipment; choices are: 0, No, N, 1, Yes, Y. **Note**: Defaults to “Y” (Yes, 1) unless otherwise specified.
string	ActivityCode	("Activity Code", dtString, Maximum Length 5)
		From a lookup list, select the type of inspection activity used to monitor the TML. Examples: UT, RT.
decimal?	ArbMinimum	("Eng Standard Thick", dtDouble, Unit-Based Field)
		The engineering standard thickness for the TML. This is normally a corporate engineering standard thickness value. This can be obtained from a linked section from the equipment Static Data. **Note**: When a Section ID from the owning equipment’s static data is populated for a TML, the engineering standard thickness will be taken from the identified section; any values entered within a call will be excluded and logged as such.
string	CircuitEquipmentID	("Circuit ID", dtString, Maximum Length 15, FK (StatHdr.EquipmentID))

<table>
<tr><td></td><td></td><td>Select the circuit number from the equipment static data where the TML is located. The list is built from the circuits that are linked to the equipment that owns the TML.
Limitation: Circuit must be linked to the TML’s owning equipment. A TML may only belong to one (1) circuit.</td></tr>
<tr><td rowspan="2">bool?</td><td rowspan="2">ConfinedSpace</td><td>("Confined Space", dtInteger)</td></tr>
<tr><td>Indicate whether the TML is located in a confined space; choices are: 0, No, N, 1, Yes, Y.</td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">CorrAllow</td><td>("Corr Allow", dtDouble, Unit-Based Field)</td></tr>
<tr><td>The given corrosion allowance for the TML. This can be obtained from the linked Section from the equipment Static Data.
Note: When a Section ID from the owning equipment’s static data is populated for a TML, the corrosion allowance will be taken from the identified section; any values entered within a call will be excluded and logged as such.</td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">CriticalBuffer</td><td>("Critical Buffer", dtDouble, Unit-Based Field)</td></tr>
<tr><td>The percent value for the buffer used to determine the critical thickness. The intent is to allow a buffer between the alarming thickness and the absolute minimum allowable thickness.</td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">DefCorrRate</td><td>("Historic CR", dtDouble, Unit-Based Field)</td></tr>
<tr><td>The corrosion rate is used to determine losses when there are no calculated minimum/average losses. For example: if the area of the TML was replaced with new pipe, a default rate can be entered based on the historical losses from the old material.</td></tr>
<tr><td rowspan="2">String</td><td rowspan="2">DMechID</td><td>("DM ID", dtString, Maximum Length 30)</td></tr>
<tr><td>From a lookup list, select the degradation mechanism identifier associated with the TML.
Note: The values are sourced from the Degradation Mechanisms Configuration library and the value must exist before updating a TML.</td></tr>
<tr><td rowspan="2">string</td><td rowspan="2">DrawingNo</td><td>("Drawing Number", dtString, Maximum Length 250)</td></tr>
<tr><td>The electronic CAD drawing number which shows the locations of the TMLs on the equipment.</td></tr>
<tr><td rowspan="2">TG29</td><td rowspan="2">Grid</td><td>("TML Type", dtInteger, Required Field)</td></tr>
<tr><td>Select the type of TML. Choices are: Point (calculates Min and Avg), Grid (positional calculations), Scan (user-entered Min and Avg)
Note: The TML Type controls the number of columns and rows for a TML, per the following rules:
<table>
<tr><th>TML Type</th><th>Rows</th><th>Columns</th></tr>
<tr><td>Point</td><td>1</td><td>>=1 and <= 99</td></tr>
<tr><td>Grid</td><td>>=1 and <= 99</td><td>>=1 and <= 99</td></tr>
<tr><td>Scan</td><td>1</td><td>2</td></tr>
</table></td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">GridAxial</td><td>("Axial Spacing", dtDouble)</td></tr>
<tr><td>Enter the axial spacing for the grid format of the TML.
Note: This is only updatable when the TML Type (Grid) is set to "Grid" and the Uniform Grid (GridUniform) is set to "No".</td></tr>
<tr><td>decimal?</td><td>GridCircumf</td><td>("Circumferential Spacing", dtDouble)
Enter the circumferential spacing for the grid format of the TML.</td></tr>
</table>

		Enter the circumferential spacing for the grid format of the TML. Note: This is only updatable when the TML Type (Grid) is set to "Grid" and the Uniform Grid (GridUniform) is set to "No".
int?	GridColumns	("Column Count", dtInteger)
		The number of columns / number of readings to be typically taken for a TML. In grid format this represents the number of columns and in non-grid this represents the maximum number of readings.
int?	GridRows	("Row Count", dtInteger)
		The number of rows to be taken in grid format for a TML. In non-grid format, this is not supported.
int?	GridUniform	("Uniform Grid", dtInteger, Default Value 1)
		Indicate whether the spacing is uniform in the grid format for the TML; choices are: 0, No, N, 1, Yes, Y.
int?	InspRiskCode	("IRC", dtInteger)
		The inspection risk code for the TML from the Inspection Risk Codes lookup data. Based on the stream setting, the restricted interval and RL factor are set. **Note**: If the RBI module is active, the RC ID must be populated to set the Inspection Risk Code. **Limitation**: If a risk assessment has been applied to a TML within the Visions Client, a message will be logged within the UpdateTML call identifying that the Inspection Risk Code cannot be updated.
bool?	InsulRemoval	("Insulation Removal", dtInteger)
		Indicate whether insulation must be removed to access the TML; choices are: 0, No, N, 1, Yes, Y.
string	ISONum	("ISO Number", dtString, Maximum Length 250)
		Enter the isometric drawing number that contains the TML.
bool?	LadderReqrd	("Ladder Required", dtInteger)
		Indicate whether a ladder is required to access the TML; choices are: 0, No, N, 1, Yes, Y.
bool?	Manlift	("Manlift", dtInteger)
		Indicate whether a manlift is required to access the TML; choices are: 0, No, N, 1, Yes, Y.
string	MaterialID	("Material ID", dtString, Maximum Length 15)
		A code determined by the User. Usually indicates the basic material and the year of the Code of reference. For example, A105/98 would be for SA-105 material from the 1998 ASME Codes. Note: The values are sourced from the Material Code Lookup Data and the value must exist before updating a TML.
TG17?	MinimumType	("Minimum Type", dtInteger)

<table>
<tr><td></td><td></td><td>Indicates how the minimum thickness was entered or calculated. Choices are:
<table>
<tr><th></th><th>Definition</th></tr>
<tr><td>NCA</td><td>Indicates that the Nominal - Corrosion Allowance was used to calculate the Minimum Thickness.</td></tr>
<tr><td>Other</td><td>Indicates that an externally calculated, typically an engineering standard, and the Minimum Thickness was user entered.</td></tr>
<tr><td>t-Min Code</td><td>Indicates that the Minimum Thickness was user entered.</td></tr>
<tr><td>t-Min Visions</td><td>Identifies that the Visions t-Min calculator was used to calculate Minimum Thickness.
<b>Limitation</b>: This option is not updateable and only available in the Visions Enterprise Client application when a Section is linked to a TML and the Section used the tMin calculator.</td></tr>
</table></td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">MinThick</td><td>("Min Thick", dtDouble, Unit-Based Field)</td></tr>
<tr><td>The minimum thickness as determined by NCA, ENGS or Code. If it is determined by NCA and the nominal and CA are entered, this minimum value is calculated automatically. Otherwise it is entered manually.
<b>Note</b>: When a Section ID from the owning equipment’s static data is populated for a TML, the minimum thickness will be taken from the identified section; any values entered within a call will be excluded and logged as such.
<b>Note</b>: If the Minimum Type is set to NCA and a minimum thickness value is provided, the calculated value will override the populated value.</td></tr>
<tr><td rowspan="2">DateTime?</td><td rowspan="2">NextInspDate</td><td>("Next Insp Date Due", dtDate)</td></tr>
<tr><td>Originally set to the date calculated based on the Restricted Interval or metal loss rates (remaining life). This date can be manually changed to match scheduled work.</td></tr>
<tr><td rowspan="2">string</td><td rowspan="2">NextInspReason</td><td>("Overwrite Reason", dtString, Maximum Length 25)</td></tr>
<tr><td>From a lookup list, select the reason for changing the default Next Inspection Due date.</td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">NomThick</td><td>("Nominal", dtDouble, Unit-Based Field)</td></tr>
<tr><td>The nominal wall thickness where the TML is located. This can be obtained from Section Design of the equipment.
<b>Note</b>: When a Section ID from the owning equipment’s static data is populated for a TML, the nominal thickness will be taken from the identified section; any values entered within a call will be excluded and logged as such.</td></tr>
<tr><td rowspan="2">TG3?</td><td rowspan="2">OnOffFlag</td><td>("Stream", dtInteger)</td></tr>
<tr><td>Set the stream that identifies the process stream condition when the TML was measured; choices are:
<table>
<tr><th>Stream</th><th>Definition</th></tr>
<tr><td>Internal</td><td>the TML is located and measured internally within its owning equipment</td></tr>
<tr><td>External (Off)</td><td>the TML is located externally and measured while its owning equipment is off stream</td></tr>
<tr><td>External (On)</td><td>the TML is located externally and measured while its owning equipment is on stream</td></tr>
</table>
<b>Note</b>: The Stream controls and validates the Restricted Interval and RL Factor, using the appropriate Internal or External values.</td></tr>
</table>

string	OtherAccess	("Other Access", dtString, Maximum Length 15)
		From a lookup list, select any additional type of access required for the TML.
string	PointClassification	("TML Classification", dtString, Maximum Length 5)
		From a lookup list, select the user classification for the part of the equipment where the TML is located. Examples: SH for shell, HD for head, BT for boot.
string	PointDescription	("TML Description", dtString, Maximum Length 200)
		The detailed description for the TML.
int?	RAID	("RC ID", dtInteger, Read-Only Field)
		Identifies the risk assessment id number set when either an IRC is selected for the TML or a risk assessment is applied to the TML. **Note**: If the RBI module is active and a risk assessment is not applied to the TML, the value will be validated as an active and completed risk configuration. **Limitation**: If a risk assessment has been applied to a TML within the Visions Client, a message will be logged within the UpdateTML call identifying that the RC ID cannot be updated.
decimal?	RestrictedInterval	("Restricted Interval", dtDouble)
		The restricted interval, in months, for the TML; used in the calculations for the next inspection date. This is initially defaulted to the RI for the selected IRC but can be over-ridden by a user. **Note**: A value of a whole number greater than or equal to 1 represents the interval in months, and a fractional number between 0.01 and 0.30 represents the interval in days.
decimal?	RLFactor	("RL Factor", dtDouble)
		The remaining life factor for the TML; used in the calculations for the expiration date. This is initially defaulted to the RL Factor for the selected IRC but can be over-ridden by a user. **Note**: A fractional number between 0 and 1; allowing 2 decimal places.
bool?	ScaffReqrd	("Scaffold Required", dtInteger)
		Indicate whether scaffolding is required to access the TML; choices are: 0, No, N, 1, Yes, Y.
string	SectionID	("Section ID", dtString, Maximum Length 35)
		Select the Section ID, Line Number or Nozzle ID from the equipment sections, nozzles, or line data identifying where the TML is located; used to set the nominal, corrosion allowance, minimum, and engineering standard thicknesses, and minimum type values from the TML's owning equipment static data.
string	Spec	("Material (Orig)", dtString, Read-Only Field, Maximum Length 15)
		Identifies the type of material for the applied section of the TML. This is defaulted based on the selected section, nozzle, or line detail.
int?	YearInstalled	("Year Installed", dtInteger)
		The year the TML was installed on the equipment.

Foreign Reference		
Table Name	**Call Value**	**Definition**
ReadGrid	lstReadGrid	Grid Readings - test dates and readings for Grid TML Types

ReadIn	lstReadIn	Readings - test dates and readings for Point and Scan TML Types
ReadNotes	lstReadNotes	Reading Notes - test dates and notes
GeoLoc	rfLocGUID	Geolocation Coordinates

GeoLoc (rfLocGUID)

Geolocation Coordinates: the centralized original GPS and linear coordinates for equipment, work, and trending.

Limitation: The fields listed below are based on the Plant to which the equipment belongs. If the plant is flagged as a Pipeline Plant, then the Linear coordinates are available; otherwise only the original GPS coordinates are available.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	Desc_End	("Description - End", dtString, Maximum Length 100) Enter a description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100) Enter a description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the ending milepost distance (chainage). The lookup list is extracted from the PL_SegMarkers Reference Markers. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). The lookup list is extracted from the PL_SegMarkers Reference Markers. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field) The ending chainage or linear location. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field) The starting chainage or linear location. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger) From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage. Choices are: Left, Right. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger)

		From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage. Choices are: Left, Right. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field) The original ending GPS Longitude (X) location coordinate point.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field) The original starting GPS Longitude (X) location coordinate point.
decimal?	Y_EndOrig	("Latitude - End", dtDouble, Unit-Based Field) The original ending GPS Latitude (Y) location coordinate point.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field) The original starting GPS Latitude (Y) location coordinate point.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field) The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - End", dtDouble, Unit-Based Field) The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign References		
Table Name	**Call Value**	**Definition**
EquipPoint	lstEquipPoint	Master TMLs (Thickness Monitoring Locations)

ReadGrid (lstReadGrid)

Grid Reading Test Dates: the test dates and associated information for TMLs characterized as Grid; represents the Test Dates grid in the TML Trending detail.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
DateTime	**TestDate**	("Test Date", dtDate, Required Field, FK (ReadGridDetail.TestDate)) Set a date for each grid-style reading. **Note**: This is a unique identifier for each test date within a TML.
string	**TMLID**	("TML ID", dtString, Required Field, Read-Only Field, Maximum Length 15) The identifier that uniquely identifies the TML for a grid-style reading.
string	ActivityCode	("Activity Code", dtString, Maximum Length 5)

<table>
<tr><td></td><td></td><td>From a lookup list, select the type of inspection activity used to perform the reading for a TML.</td></tr>
<tr><td rowspan="2">TG86?</td><td rowspan="2">BaselineFlag</td><td>("Baseline", dtInteger, Default Value None)</td></tr>
<tr><td>Identify whether the reading test date is a baseline measurement; choices are:
<table>
<tr><th>Baseline</th><th>Definition</th></tr>
<tr><td>None</td><td>The thickness measurements for the test date are not a baseline survey.</td></tr>
<tr><td>Actual</td><td>The thickness measurements for the test date are actual, surveyed readings.</td></tr>
<tr><td>Nominal</td><td>The thickness measurements for the test date are based on nominal thickness value(s).</td></tr>
</table></td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">CurrentTemp</td><td>("Current Temp", dtDouble, Unit-Based Field)</td></tr>
<tr><td>The actual temperature of the metal at the TML at the time the thickness value was obtained.</td></tr>
<tr><td rowspan="2">string</td><td rowspan="2">Inspector</td><td>("Inspector", dtString, Maximum Length 35)</td></tr>
<tr><td>The name of the inspector who obtained the thickness measurement.</td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">LongLoss</td><td>("Long Loss", dtDouble, Read-Only Field, Unit-Based Field)</td></tr>
<tr><td>Displays the greatest, or worst case, long loss of the all grid coordinates for a TML test date.
Limitation: This is a calculated value and cannot be updated.</td></tr>
<tr><td rowspan="2">TG3?</td><td rowspan="2">OnOffFlag</td><td>("Stream", dtInteger)</td></tr>
<tr><td>Set the stream that identifies the process stream condition when the TML was measured; choices are:
<table>
<tr><th>Stream</th><th>Definition</th></tr>
<tr><td>Internal</td><td>the TML is located and measured internally within its owning equipment</td></tr>
<tr><td>External (Off)</td><td>the TML is located externally and measured while its owning equipment is off stream</td></tr>
<tr><td>External (On)</td><td>the TML is located externally and measured while its owning equipment is on stream</td></tr>
</table></td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">ShortLoss</td><td>("Short Loss", dtDouble, Read-Only Field, Unit-Based Field)</td></tr>
<tr><td>Displays the greatest, or worst case, short loss of the all grid coordinates for a TML test date.
Limitation: This is a calculated value and cannot be updated.</td></tr>
</table>

Foreign Reference		
Table Name	**Call Value**	**Definition**
ReadGrid	lstReadGrid	Grid Readings - test dates and readings for Grid TML Types
EquipPoint	rfEquipPoint	Master table for TML Trending; TML ID details

ReadGridDetail (lstReadGridDetail)

Grid Readings: the individual readings for TMLs categorized as Grid; represents the Readings grid in the TML Trending detail.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)

DateTime	**TestDate**	("Test Date", dtDate, Required Field, FK (ReadGridDetail.TestDate)) Set a date for each grid-style reading.
int	**RdColID**	("Column ID", dtInteger, Required Field) A system generated counter that uniquely identifies the column position for a reading within a grid TML. Note: The ReadingsCSV array parameter is used for reading addition or editing. This field is provided for information only.
int	**RdRowID**	("Row ID", dtInteger, Required Field) A system generated counter that uniquely identifies the row position for reading within a grid TML. Note: The ReadingsCSV array parameter is used for reading addition or editing. This field is provided for information only.
string	**TMLID**	("TML ID", dtString, Required Field, Read-Only Field, Maximum Length 15) The identifier that uniquely identifies the TML for a grid-style reading.
decimal?	LongLoss	("Long Loss", dtDouble, Read-Only Field, Unit-Based Field) Displays the greatest, or worst case, long loss for a specific grid coordinate. **Limitation**: This is a calculated value and cannot be updated.
decimal?	Reading	("Reading", dtDouble, Required Field, Unit-Based Field) The thickness measurement for the identified column/row cell within a grid TML. Note: Represented by the ReadingsCSV array parameter.
decimal?	ShortLoss	("Short Loss", dtDouble, Read-Only Field, Unit-Based Field) Displays the greatest, or worst case, short loss for a specific grid coordinate. **Limitation**: This is a calculated value and cannot be updated.

Foreign Reference		
Table Name	**Call Value**	**Definition**
ReadGrid	rfReadGrid	Readings - Test Date information for Grid TML Types

Readin (lstReadIn)

Point/Scan Reading Test Dates: the test dates and associated information for TMLs characterized as Point or Scan; represents the Test Dates grid in the TML Trending detail.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
DateTime	**TestDate**	("Test Date", dtDate, Required Field, FK (ReadDetail.TestDate)) Set a date for each point/scan type reading. **Note**: This is a unique identifier for each test date within a TML.
string	**TMLID**	("TML ID", dtString, Required Field, Read-Only Field, Maximum Length 15) The identifier that uniquely identifies the TML for a point/scan type reading.
string	ActivityCode	("Activity Code", dtString, Maximum Length 5) From a lookup list, select the type of inspection activity used to perform the reading for a TML.

<table>
<tr><td rowspan="2">TG86?</td><td rowspan="2">BaselineFlag</td><td>("Baseline", dtInteger, Default Value None)</td></tr>
<tr><td>Identify whether the reading test date is a baseline measurement; choices are:
<table>
<tr><th>Baseline</th><th>Definition</th></tr>
<tr><td>None</td><td>The thickness measurements for the test date are not a baseline survey.</td></tr>
<tr><td>Actual</td><td>The thickness measurements for the test date are actual, surveyed readings.</td></tr>
<tr><td>Nominal</td><td>The thickness measurements for the test date are based on nominal thickness value(s).</td></tr>
</table></td></tr>
<tr><td rowspan="2">decimal?</td><td rowspan="2">CurrentTemp</td><td>("Current Temp", dtDouble, Unit-Based Field)</td></tr>
<tr><td>The actual temperature of the metal at the TML at the time the thickness value was obtained.</td></tr>
<tr><td rowspan="2">string</td><td rowspan="2">Inspector</td><td>("Inspector", dtString, Maximum Length 35)</td></tr>
<tr><td>The name of the inspector who obtained the thickness value.</td></tr>
<tr><td rowspan="2">TG3?</td><td rowspan="2">OnOffFlag</td><td>("Stream", dtInteger)</td></tr>
<tr><td>Select the stream that identifies the process stream condition when the TML was measured; choices are:
<table>
<tr><th>Stream</th><th>Definition</th></tr>
<tr><td>Internal</td><td>the TML is located and measured internally within its owning equipment</td></tr>
<tr><td>External (Off)</td><td>the TML is located externally and measured while its owning equipment is off stream</td></tr>
<tr><td>External (On)</td><td>the TML is located externally and measured while its owning equipment is on stream</td></tr>
</table></td></tr>
</table>

Foreign Reference		
Table Name	**Call Value**	**Definition**
ReadGrid	lstReadGrid	Grid Readings - test dates and readings for Grid TML Types
EquipPoint	rfEquipPoint	Master table for TML Trending; TML ID details

ReadDetail (lstReadDetail)

Point/Scan Readings: the individual readings for TMLs categorized as Point or Scan; represents the Readings grid in the TML Trending detail.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
DateTime	**TestDate**	("Test Date", dtDate, Required Field, Read-Only Field, FK (ReadIn.TestDate))
		Set a date for each point/scan type reading.
string	**TMLID**	("TML ID", dtString, Required Field, Read-Only Field, Maximum Length 15, FK (ReadIn.TMLID))
		The identifier that uniquely identifies the TML for a point/scan type reading.
int	**ReadCounter**	("Reading Counter", dtInteger, Required Field)

		A system generated counter that uniquely identifies the column position for a reading within a point/scan type TML. Note: The ReadingsCSV array parameter is used for reading addition or editing. This field is provided for information only.
decimal	Reading	("Reading", dtDouble, Required Field, Unit-Based Field) The thickness measurement for the identified reading counter within a point/scan TML. Note: Represented by the ReadingsCSV array parameter.

Foreign Reference		
Table Name	**Call Value**	**Definition**
ReadIn	rfReadIn	Readings - Test Date information for Point and Scan TML Types

ReadNotes (lstReadNotes)

TML Notes: the individual notes for TMLs; represents the Notes grid in the TML Trending detail. Dates for TML notes may or may not align with the reading test dates for the TML to allow for notes to be added where thicknesses are not measured.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
DateTime	**TestDate**	("Test Date", dtDate, Required Field, Read-Only Field, FK (ReadIn.TestDate)) Set the test date for each reading note. **Note**: The test date may different or the same as a reading test date for a TML.
string	**TMLID**	("TML ID", dtString, Required Field, Read-Only Field, Maximum Length 15, FK (ReadIn.TMLID)) The identifier that uniquely identifies the TML for a reading note.
string	Notes	("Notes", dtString, Maximum Length 4000) Any notes or comments for the notes test date of a TML.
string	NotesCategory	("Notes Category", dtString, Maximum Length 25) From a lookup list, select the category for the notes.

Foreign Reference		
Table Name	**Call Value**	**Definition**
EquipPoint	rfEquipPoint	Master table for TML Trending; TML ID details.

Call Examples

Similar to the Asset API, a debug flag may be included in a call to echo the data in the response message for troubleshooting.

QueryTML

An example which includes the new Notes fields and the Long/ShortLossCSV fields.

Call without Debug Information

```
<!-- You can use QueryTML by specifying a specific TML using the hardcoded EquipNum,
    PlantNum, EquipType and TMLID parameters; without a TMLID specified data for all the
    TMLs specified will be returned. If you do not specify any criteria, all the TMLs in
    the database will be returned. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:met="http://metegrity.com"
    xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
 <soapenv:Header/>
 <soapenv:Body>
   <met:QueryTML>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-105A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:pageSize>20</met:pageSize>
    <met:pageNum>0</met:pageNum>
   </met:QueryTML>
 </soapenv:Body>
</soapenv:Envelope>

<!-- Alternatively, you can explicitly name up to four fields you want to select TMLs using;
    these can be any of the fields that are returned from the query. Using this syntax,
    you can specify more than one permissible value for each field, allowing you to retrieve
    a list of specific TMLs with multiple TMLIDs or TMLs from a list of different Equipment.
    Any given TML will be returned if, for each specified field, its value matches one of
    the specified values. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:met="http://metegrity.com"
    xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
 <soapenv:Header/>
 <soapenv:Body>
    <met:QueryTML>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:FieldName1>EquipPoint.TMLID</met:FieldName1>
    <met:FieldValues1>
       <arr:string>V105-020</arr:string>
    </met:FieldValues1>
    <met:pageSize>20</met:pageSize>
    <met:pageNum>0</met:pageNum>
    </met:QueryTML>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!--   QueryTML returns results in an array. The first row gives the column headers for
    all the other rows. The specific columns included can be customized using the ColumnSet
    string. A number of data rows equal to pageSize will be returned, starting at the
    (pageNum x pageSize)th row, and the moreLeft indicator will indicate whether this was
    the final page.  QueryTML returns (only) the most recent set of readings for each TML.
    They are included as a single column, with the readings in sequential order separated
    by semicolons; as such, a TML with a 3x3 grid and no entered readings will have a
    sequence of 8 semicolons here (i.e., nine zero-length strings separated by semicolons).
    QueryTML also returns the LongLoss and ShortLess for each Reading location, in a similar
    format to the actual readings.  The most recent notes for each TML are included as
    well. Any RTF formatting is stripped from the notes content to render them in plain
    text. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
<QueryTMLResponse xmlns="http://metegrity.com">
 <QueryTMLResult>true</QueryTMLResult>
 <OutList xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
   xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
   <a:ArrayOfstring>
    <a:string>Classification</a:string>
    <a:string>Plant ID</a:string>
    <a:string>Equipment Type</a:string>
    <a:string>Equipment Number</a:string>
    <a:string>TML ID</a:string>
    <a:string>TML Type</a:string>
    <a:string>Section ID</a:string>
    <a:string>TML Description</a:string>
    <a:string>Column Count</a:string>
    <a:string>Row Count</a:string>
    <a:string>Circuit ID</a:string>
    <a:string>Location</a:string>
    <a:string>Associated Major</a:string>
    <a:string>TML Description</a:string>
    <a:string>Corr Circuit ID</a:string>
    <a:string>Section ID</a:string>
    <a:string>Year Installed</a:string>
    <a:string>Nominal</a:string>
    <a:string>Corr Allow</a:string>
    <a:string>Min Thick</a:string>
    <a:string>Critical Thick</a:string>
    <a:string>Stream</a:string>
    <a:string>Active</a:string>
    <a:string>Insulation Removal</a:string>
    <a:string>Scaffold Required</a:string>
    <a:string>Manlift</a:string>
    <a:string>Ladder Required</a:string>
    <a:string>Confined Space</a:string>
    <a:string>Other Access</a:string>
    <a:string>Activity Code</a:string>
    <a:string>Min Short Loss</a:string>
    <a:string>Min Long Loss</a:string>
    <a:string>Avg Short Loss</a:string>
    <a:string>Avg Long Loss</a:string>
    <a:string>IRC</a:string>
    <a:string>RL Factor</a:string>
    <a:string>Restricted Interval</a:string>
    <a:string>TML Classification</a:string>
    <a:string>Next Insp Date Due</a:string>
    <a:string>Next Insp Date Calculated</a:string>
    <a:string>Last Reading Date</a:string>
    <a:string>Last Read Min Thick</a:string>
    <a:string>Last Read Avg Thick</a:string>
    <a:string>Expiration Date</a:string>
    <a:string>ReadingsCSV</a:string>
    <a:string>LongLossCSV</a:string>
    <a:string>ShortLossCSV</a:string>
    <a:string>Last Notes Date</a:string>
    <a:string>Notes Category</a:string>
    <a:string>Last Notes Text</a:string>
   </a:ArrayOfstring>
   <a:ArrayOfstring>
    <a:string>Equipment</a:string>
    <a:string>Plant 1</a:string>
    <a:string>Vessel</a:string>
    <a:string>V-105A</a:string>
    <a:string>V-105A-010</a:string>
    <a:string>Grid</a:string>
    <a:string i:nil="true"/>
    <a:string>Shell wall</a:string>
    <a:string>3</a:string>
    <a:string>3</a:string>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string>Testing API</a:string>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string>Internal</a:string>
    <a:string>False</a:string>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string>False</a:string>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string>Civil</a:string>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string>EFR</a:string>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string>08/07/2015</a:string>
    <a:string>1.1</a:string>
    <a:string>5.5</a:string>
    <a:string i:nil="true"/>
    <a:string>1.1;2.2;3.3;4.4;5.5;6.6;7.7;8.8;9.9</a:string>
    <a:string>;;;;;;;;</a:string>
    <a:string>;;;;;;;;</a:string>
    <a:string>08/07/2015</a:string>
    <a:string>Prep Note</a:string>
    <a:string>This text is in RTF.</a:string>
   </a:ArrayOfstring>
 </OutList>
 <moreLeft>false</moreLeft>
 <ErrorMsg>Operation successful.</ErrorMsg>
</QueryTMLResponse>
```

```
 </s:Body>
</s:Envelope>
```

CreateTML

Call without Debug Information

```
<!-- The CreateTML function creates a new TML for an existing piece of equipment. If the
    TMLID already exists in that equipment, the call will fail. As many fields as are
    desired can be specified in the Values section, but a newly created TML must have a
    PlantNum, EquipNum, EquipType, TMLID, TMLType, RowCount and ColumnCount. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:CreateTML>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-105A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:TMLID>V-105A-025</met:TMLID>
    <met:TMLDesc>Testing API</met:TMLDesc>
    <met:TMLType>Point</met:TMLType>
    <met:RowCount>1</met:RowCount>
    <met:ColumnCount>3</met:ColumnCount>
    <met:Values>
        [Values]
        [Value field='ActivityCode']RT[/Value]
        [Value field='OnOffFlag']Internal[/Value]
        [Value field='ActiveFlag']false[/Value]
        [/Values]
    </met:Values>
   </met:CreateTML>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The CreateTML response message simply indicates whether the action was successful and
    returns any relevant error messages. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <CreateTMLResponse xmlns="http://metegrity.com">
    <CreateTMLResult>true</CreateTMLResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </CreateTMLResponse>
 </s:Body>
</s:Envelope>
```

CloneTML

Call without Debug Information

```
<!-- The CloneTML function creates a new TML for an existing piece of equipment. If the
    TMLID already exists in that equipment, the call will fail.  It copies the TML identified
    by the four “Src” parameters to each piece of equipment and new TMLID identified by
    the four “Dest” parameters. These are parallel arrays, so DestTMLID[3] will be created
    in the equipment identified by DestEquipNum[3], DestPlantNum[3] and DestEquipType[3];
    the equipment identifiers can be repeated for each new TML to be cloned from the
    source.-->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com"
   xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
 <soapenv:Header/>
 <soapenv:Body>
   <met:CloneTML>
      <met:SiteName>[Visions SiteName]</met:SiteName>
      <met:UserName>[Visions UserName]</met:UserName>
      <met:Password>[Visions Password]</met:Password>
      <met:SrcEquipNum>P101-12-6-F</met:SrcEquipNum>
      <met:SrcPlantNum>Plant 1</met:SrcPlantNum>
      <met:SrcEquipType>Pipe</met:SrcEquipType>
      <met:SrcTMLID>P101-005</met:SrcTMLID>
      <met:DestEquipNum>
         <arr:string>V-1101A</arr:string>
         <arr:string>V-1101A</arr:string>
         <arr:string>V-1101A</arr:string>
         <arr:string>V-1101A</arr:string>
      </met:DestEquipNum>
      <met:DestPlantNum>
         <arr:string>U-1100 - Feed Dryers</arr:string>
         <arr:string>U-1100 - Feed Dryers</arr:string>
         <arr:string>U-1100 - Feed Dryers</arr:string>
         <arr:string>U-1100 - Feed Dryers</arr:string>
      </met:DestPlantNum>
      <met:DestEquipType>
         <arr:string>Vessel</arr:string>
         <arr:string>Vessel</arr:string>
         <arr:string>Vessel</arr:string>
         <arr:string>Vessel</arr:string>
      </met:DestEquipType>
      <met:DestTMLID>
         <arr:string>V-1101A-010</arr:string>
         <arr:string>V-1101A-011</arr:string>
         <arr:string>V-1101A-012</arr:string>
         <arr:string>V-1101A-013</arr:string>
      </met:DestTMLID>
   </met:CloneTML>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The CloneTML response message simply indicates whether the action was successful and
    returns any relevant error messages. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Body>
      <CloneTMLResponse xmlns="http://metegrity.com">
         <CloneTMLResult>true</CloneTMLResult>
         <ErrorMsg>Operation successful.</ErrorMsg>
      </CloneTMLResponse>
   </s:Body>
</s:Envelope>
```

CloneTMLWithDates

Call without Debug Information

```
<!-- The CloneTMLWithDates function creates a new TML for an existing piece of equipment
    setting the next inspection due date. If the TMLID already exists in that equipment,
    the call will fail.  It copies the TML identified by the four “Src” parameters to each
    piece of equipment and new TMLID identified by the four “Dest” parameters.  It sets
    the next inspection due date based on the identified date format "dateFmt". These are
    parallel arrays, so DestTMLID[3] will be created in the equipment identified by
    DestEquipNum[3], DestPlantNum[3], and DestEquipType[3] and setting the dueDates[3];
    the equipment and date identifiers can be repeated for each new TML to be cloned from
    the source.-->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:met="http://metegrity.com"
    xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
   <soapenv:Header/>
   <soapenv:Body>
      <met:CloneTMLWithDates>
         <met:SiteName>[Visions SiteName]</met:SiteName>
         <met:UserName>[Visions UserName]</met:UserName>
         <met:Password>[Visions Password]</met:Password>
         <met:SrcEquipNum>P101-12-6-F</met:SrcEquipNum>
         <met:SrcPlantNum>Plant 1</met:SrcPlantNum>
         <met:SrcEquipType>Pipe</met:SrcEquipType>
         <met:SrcTMLID>P101-005</met:SrcTMLID>
         <met:DestEquipNum>
            <arr:string>P101-12-6-F</arr:string>
            <arr:string>P101-12-6-F</arr:string>
            <arr:string>P101-12-6-F</arr:string>
            <arr:string>P101-12-6-F</arr:string>
         </met:DestEquipNum>
         <met:DestPlantNum>
            <arr:string>Plant 1</arr:string>
            <arr:string>Plant 1</arr:string>
            <arr:string>Plant 1</arr:string>
            <arr:string>Plant 1</arr:string>
         </met:DestPlantNum>
         <met:DestEquipType>
            <arr:string>Pipe</arr:string>
            <arr:string>Pipe</arr:string>
            <arr:string>Pipe</arr:string>
            <arr:string>Pipe</arr:string>
         </met:DestEquipType>
         <met:DestTMLID>
            <arr:string>P101-010</arr:string>
            <arr:string>P101-015</arr:string>
            <arr:string>P101-020</arr:string>
            <arr:string>P101-025</arr:string>
         </met:DestTMLID>
         <met:dateFmt>MM/dd/yyyy</met:dateFmt>
         <met:dueDates>
            <arr:string>07/08/2020</arr:string>
            <arr:string>07/08/2020</arr:string>
            <arr:string>07/08/2020</arr:string>
            <arr:string>07/08/2020</arr:string>
         </met:dueDates>
      </met:CloneTMLWithDates>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The CloneTMLWithDates response message simply indicates whether the action was
    successful and returns any relevant error messages. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Body>
      <CloneTMLWithDatesResponse xmlns="http://metegrity.com">
         <CloneTMLWithDatesResult>true</CloneTMLWithDatesResult>
         <ErrorMsg>Operation successful.</ErrorMsg>
      </CloneTMLWithDatesResponse>
   </s:Body>
</s:Envelope>
```

UpdateTML

Call with Debug Information

```
<!-- UpdateTML is used to update values for an existing TML; the call fails if the TML does
    not already exist. Several fields, such as TMLType and RowCount, cannot be changed once
    a TML has been created. Others, such as RCID, can only be changed if they haven't
    already been set. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
    <met:UpdateTML>
     <met:SiteName>[Visions SiteName]</met:SiteName>
     <met:UserName>[Visions UserName]</met:UserName>
     <met:Password>[Visions Password]</met:Password>
     <met:EquipNum>V-105A</met:EquipNum>
     <met:PlantNum>Plant 1</met:PlantNum>
     <met:EquipType>Vessel</met:EquipType>
     <met:TMLID>V105A-025</met:TMLID>
     <met:Values>
         [Values]
         [DebugInfo /]
         [Value field='Manlift']false[/Value]
         [Value field='PointClassification']SHL[/Value]
         [/Values]
     </met:Values>
    </met:UpdateTML>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The Values section for our example UpdateTML call included the "[DebugInfo /]" node.
    This causes the data actually being written to be echoed back in the response message
    as a troubleshooting measure so programmers can see what is happening, as shown below.
    -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
    <UpdateTMLResponse xmlns="http://metegrity.com">
     <UpdateTMLResult>true</UpdateTMLResult>
     <ErrorMsg>Operation successful.
        ----
```

```
       [?xml version="1.0" encoding="utf-16"?]
       [DataArray count="1" p1:xsi="&amp;quot;http://www.w3.org/2001/XMLSchema-
       instance&amp;quot;" xmlns:p1="xmlns"]
       [DataClass p3:type="Metegrity.Data.EquipPoint" xsi:type="EquipPoint" ID="1000"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:p3="xsi"]
       [TF]*[/TF]
       [EquipmentID]V5DI00000000VOS[/EquipmentID]
       [TMLID]V-105A-020[/TMLID]
       [PointDescription]Shell Wall thickness grid[/PointDescription]
       [OnOffFlag]12[/OnOffFlag]
       [ActiveFlag]false[/ActiveFlag]
       [Manlift]false[/Manlift]
       [ActivityCode]Civil[/ActivityCode]
       [DefCorrRate]5.25[/DefCorrRate]
       [LastReadingDate]635719104000000000[/LastReadingDate]
       [LastReadMinThick]2[/LastReadMinThick]
       [LastReadAveThick]6.3333[/LastReadAveThick]
       [PointClassification]EFR[/PointClassification]
       [CriticalBuffer]10.25[/CriticalBuffer]
       [MinimumType]142[/MinimumType]
       [Grid]191[/Grid]
       [GridRows]3[/GridRows]
       [GridColumns]3[/GridColumns]
       [LastUpdate]635938143810000000[/LastUpdate]
       [UpdatedBy]JULIAN[/UpdatedBy]
       [/DataClass]
       [/DataArray]
       ----</ErrorMsg>
   </UpdateTMLResponse>
 </s:Body>
</s:Envelope>
```

Call with Field Restriction

```
<!-- Visions' calculated fields cannot be written to using the TML API, as their values
    are determined by internal calculations. This call-response pair shows an effort to do
    so being rejected. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:UpdateTML>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-105A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:TMLID>V105A-025</met:TMLID>
    <met:Values>
        [Values]
       [DebugInfo /]
        [Value field='Manlift']false[/Value]
        [Value field='PointClassification']EFR[/Value]
        [Value field='AveRegLoss']77[/Value]
        [/Values]
    </met:Values>
   </met:UpdateTML>
 </soapenv:Body>
```

```
</soapenv:Envelope>
```

Response

```
<!-- The response message indicates the field(s) that do not allow editing. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <UpdateTMLResponse xmlns="http://metegrity.com">
    <UpdateTMLResult>false</UpdateTMLResult>
    <ErrorMsg>API call does not permit writing to field
   'EquipPoint.AveRegLoss'.</ErrorMsg>
   </UpdateTMLResponse>
 </s:Body>
</s:Envelope>
```

DeleteTML

Call without Debug Information for a single record

```
<!-- The DeleteTML call deletes an entire TML and all associated data from the database.
    The syntax below shows a single TML being deleted using the explicitly named parameters.
    -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:DeleteTML>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-105A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:TMLID>V105A-020</met:TMLID>
   </met:DeleteTML>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The response message simply indicates the deletion was successful. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <DeleteTMLResponse xmlns="http://metegrity.com">
    <DeleteTMLResult>true</DeleteTMLResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </DeleteTMLResponse>
 </s:Body>
</s:Envelope>
```

Call for multiple records

```
<!-- Multiple TMLs can be deleted at once by specifying their key values in [Values] nodes
    within a [Records] node in the Values parameter XML. This example call deletes two TMLs
    with IDs NewTML6 and NewTML9. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
```

```
   <met:DeleteTML>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-105A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:Values>
        [Records]
        [Values]
        [Value field="TMLID"]V105A-005[/Value]
        [/Values]
        [Values]
        [Value field="TMLID"]V-105A-010[/Value]
        [/Values]
        [/Records]
   </met:Values>
   </met:DeleteTML>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The response messsage indicates that both TMLs were deleted. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <DeleteTMLResponse xmlns="http://metegrity.com">
    <DeleteTMLResult>true</DeleteTMLResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </DeleteTMLResponse>
 </s:Body>
</s:Envelope>
```

AddTMLReadings

Call

```
<!--   AddTMLReadings enters readings for a TML.  Readings are formatted as a concatenated
    string of floating-point values separated by semicolons.  There must be enough reading-
    positions for the number of readings in a TML (i.e., RowCount x ColumnCount), but any
    specific position can be left (null) by specifying an empty string ("") rather than a
    number.   Readings imported with AddTMLReadings are not inserted directly into the
    database.  Instead, they are placed in the TML Quarantine and are approved and imported
    using the VisLogger utility in the normal fashion. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:AddTMLReadings>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-105A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:TMLID>V105A-030</met:TMLID>
    <met:ReadingsCSV>1;2;3;4;5;6;7;8;9</met:ReadingsCSV>
    <met:TestDate>07/08/2015</met:TestDate>
    <met:dateFmt>MM/dd/yyyy</met:dateFmt>
   </met:AddTMLReadings>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The response indicates the readings were imported successfully. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <AddTMLReadingsResponse xmlns="http://metegrity.com">
    <AddTMLReadingsResult>true</AddTMLReadingsResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </AddTMLReadingsResponse>
 </s:Body>
</s:Envelope>
```

EditTMLReadings

Call

```
<!-- The EditTMLReadings call alters a specific set of existing readings for a given Test
    Date. Note that EditTMLReadings affects readings actually in the TML, not those in the
    TML Quarantine. As such, it can not be used to edit readings input with AddTMLReadings
    until they have been approved with the VisLogger utility. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:EditTMLReadings>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-105A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:TMLID>V105A-010</met:TMLID>
    <met:ReadingsCSV>1.1;2.2;3.3;4.4;5.5;6.6;7.7;8.8;9.9</met:ReadingsCSV>
    <met:TestDate>07/08/2015</met:TestDate>
    <met:dateFmt>MM/dd/yyyy</met:dateFmt>
   </met:EditTMLReadings>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The response indicates the edit was successful. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <EditTMLReadingsResponse xmlns="http://metegrity.com">
    <EditTMLReadingsResult>true</EditTMLReadingsResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </EditTMLReadingsResponse>
 </s:Body>
</s:Envelope>
```

Call with Null and Omitted Values

```
<!-- This call to EditTMLReadings demonstrates the use of a the placeholder value $NULL
    (which explicitly sets an existing reading to null, much like the similar value in
    Excel Wizard) as well as leaving specific reading positions blank (which leaves the
    reading unchanged in the database. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:EditTMLReadings>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-105A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:TMLID>V105A-015</met:TMLID>
    <met:ReadingsCSV>;$NULL;12;;$NULL;12;;$NULL;12</met:ReadingsCSV>
    <met:TestDate>07/08/2015</met:TestDate>
    <met:dateFmt>MM/dd/yyyy</met:dateFmt>
   </met:EditTMLReadings>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The response indicates the edit was successful. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <EditTMLReadingsResponse xmlns="http://metegrity.com">
    <EditTMLReadingsResult>true</EditTMLReadingsResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </EditTMLReadingsResponse>
 </s:Body>
</s:Envelope>
```

DeleteTMLReadings

Call

```
<!-- This call deletes the readings from the indicated equipment, TMLID and Test Date.-->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:DeleteTMLReadings>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-105A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:TMLID>V-105A-015</met:TMLID>
    <met:TestDate>07/08/2015</met:TestDate>
    <met:dateFmt>MM/dd/yyyy</met:dateFmt>
   </met:DeleteTMLReadings>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The deletion was successful. -->
```

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <DeleteTMLReadingsResponse xmlns="http://metegrity.com">
    <DeleteTMLReadingsResult>true</DeleteTMLReadingsResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </DeleteTMLReadingsResponse>
 </s:Body>
</s:Envelope>
```

AddTMLNotes

Call

```
<!-- The AddTMLNotes API adds the indicated Notes text to a TML for a given Test Date. The
    specified Notes Category must be already defined in Visions. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:AddTMLNotes>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-105A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:TMLID>V-105A-020</met:TMLID>
    <met:TestDate>07/08/2015</met:TestDate>
    <met:dateFmt>MM/dd/yyyy</met:dateFmt>
    <met:Category>Access</met:Category>
    <met:Notes>Sample Notes</met:Notes>
   </met:AddTMLNotes>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The API accepted the new note without issue. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <AddTMLNotesResponse xmlns="http://metegrity.com">
    <AddTMLNotesResult>true</AddTMLNotesResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </AddTMLNotesResponse>
 </s:Body>
</s:Envelope>
```

Additional Information

Visions Enterprise Help File

Lookup Data > F3 Lookup Lists

Thickness Monitoring (TML) > TML Trending

VisQuarantine > TML Review

CP Testing API

Overview

Several application programming interfaces (API) are available to allow you to perform certain actions without using the Visions Enterprise user interface (UI). This allows third party developers to use web service calls to get or update Visions data.

The Visions API for CP Testing allows you to manage cathodic protection test points; such as creating, updating, or querying CP Test Point identifiers, readings, and notes data.

Access

Navigation

The web service calls are expressed as a C# interface, with the individual calls, conventions, references, and examples documented below.

```
public interface IVisAPI {
...
}
```

Conditions

1. The CP Testing API is a separate module which must be active to use its calls.
2. Some equipment types may not own TMLs, this includes any types derived from the following supplied types: Circuit, Major, Centrifugal Pump, GP Steam Turbine, and RE Motor.
3. Security to perform calls are dependent upon the security access for the username passed into a call. For example: if you do not have Delete CP Test Point privileges, you will not be able to call the DeleteCPTP call successfully. Security functions for CP Testing are:
    a. **Open CP Testing**: controls access to the CP Testing, all calls
    b. **Add CP Test Point**: controls CP test point creation or cloning (equivalent to copy or duplicate within the Visions Client)
    c. **Update CP Test Point**: controls editing of the Interval, Definition, or Test Point ID for a test point

       Limitation: Once test dates or notes have been added to a test point, the definition may not be modified.
    d. **Delete CP Test Point**: controls deletion of a CP test point

Input

The CP Testing API provides the following calls:

- **CreateCPTP**: create the identified CP test point(s); simulates the ‘New’ action on the CP Testing Log.

  Note: Arrays are parallel, so ReadingLabels[2] applies to the same column that MaximumReadings[2] does. Specifying arrays with different lengths will result in an error. The LabelColumns parameter gives the Visions Column Counters for each CP Column, and will be populated implicitly if it is omitted.

  Security: Permission is dependent upon the ‘Add CP Test Point’ function.

- **CloneCPTP**: clone the identified CP test point(s), creating new test point(s) including the column definitions; simulates the ‘Copy’ or ‘Duplicate’ actions on the CP Testing Log

  Security: Permission is dependent upon the ‘Add CP Test Point’ function.

- **UpdateCPTP**: update an existing CP Test point, including activation and deactivation; excludes changes to readings and owning equipment

  Note: Depending on what is changed within a test point, the scheduling information for the CP Test Point and possibly its owning equipment may be impacted.

  Security: For some fields, permission is dependent upon the ‘Update CP Test Point’ function.

- **DeleteCPTP**: deletes the identified CP test point(s); simulates the ‘Delete’ action on the CP Testing Log

  Security: Permission is dependent upon the ‘Delete CP Test Point’ function.

- **QueryCPTP**: query the CP test points, returning the value(s) in a table form, including both their definitions and their latest readings; the readings are returned in a CSV string with rows appended in sequential order; each row consists of a reading for each column described in the "Column_Definitions" entry in the order the columns appear
- **AddCPTPReadings**: add new readings to a test date for CP test point(s); simulates the ‘Add Readings’ on the CP Testing detail or the CP Testing workbook mapping in the Excel Wizard

  Reference the Data Format section for details on the structure to follow a test point’s column definition.

- **EditCPTPReadings**: edit readings for the CP test point(s); either edit existing readings or add new column values to an existing test date
- **DeleteCPTPReadings**: delete a test date and all assocated readings from CP test point(s); optionally delete specific rows of readings without deleting the full test date
- **AddCPTPNotes**: add notes to the CP test point(s) for a specified test date

Calls

```
bool QueryCPTP (string SiteName, string UserName, string Password, string EquipNum,
string PlantNum, string EquipType, string CPID, string FieldName1, string[]FieldValues1,
string FieldName2, string[]FieldValues2, string FieldName3, string[]FieldValues3, string
FieldName4, string[]FieldValues4, string DateFmt, string ColumnSet, int pageSize, int
pageNum, out string[][]OutList, out bool moreLeft, out string ErrorMsg, string MapID);

bool CreateCPTP (string SiteName, string UserName, string Password, string EquipNum,
string PlantNum, string EquipType, string TestPointID, string Interval, string
IntervalUnit, int[]LabelColumns, string[]ReadingLabels, string[]MinimumReadings,
string[]MaximumReadings, string[]ReadingUnits, string Values, out string ErrorMsg, string
MapID);
```

```
bool CloneCPTP (string SiteName, string UserName, string Password, string EquipNum,
string PlantNum, string EquipType, string TestPointID, string srcEquipNum, string
srcPlantNum, string srcEquipType, string srcTestPointID, string Values, out string
ErrorMsg, string MapID);

bool UpdateCPTP (string SiteName, string UserName, string Password, string EquipNum,
string PlantNum, string EquipType, string TestPointID, string Interval, string
IntervalUnit, int[] LabelColumns, string[] ReadingLabels, string[] MinimumReadings,
string[] MaximumReadings, string[] ReadingUnits, string Values, out string ErrorMsg,
string MapID);

bool DeleteCPTP (string SiteName, string UserName, string Password, string EquipNum,
string PlantNum, string EquipType, string TestPointID, string Values, out string
ErrorMsg, string MapID);

bool AddCPTPReadings (string SiteName, string UserName, string Password, string EquipNum,
string PlantNum, string EquipType, string TestPointID, string TestDate, string Baseline,
string Stream, string Inspector, string[] Rows, string[] Columns, string[] Distance,
string[] CPState, string[] ReadingsCSV, string dateFmt, string Values, out string
ErrorMsg, string MapID);

bool EditCPTPReadings (string SiteName, string UserName, string Password, string
EquipNum, string PlantNum, string EquipType, string TestPointID, string TestDate, string
Baseline, string Stream, string Inspector, string[] Rows, string[] Columns, string[]
Distance, string[] CPState, string[] ReadingsCSV, string dateFmt, string Values, out
string ErrorMsg, string MapID);

bool DeleteCPTPReadings (string SiteName, string UserName, string Password, string
EquipNum, string PlantNum, string EquipType, string TestPointID, string TestDate,
string[] Rows, string dateFmt, string Values, out string ErrorMsg, string MapID);

bool AddCPTPNotes (string SiteName, string UserName, string Password, string EquipNum,
string PlantNum, string EquipType, string TestPointID, string TestDate, string dateFmt,
string Category, string Notes, string Values, out string ErrorMsg, string MapID);
```

Calling Conventions

The web service is a standard SOAP web service and may be consumed by anything that can consume SOAP web services.

The following conventions are specific to the CP API:

- API calls which reference a piece of equipment do so by its three (3) logical keys – PlantNum, EquipType, and EquipNum – not our internal Visions account identifier.
- API calls to interact with existing CP Test points must reference them by the logical key of TestPointID and asset identification (Plant, Equipment Type, and Equipment Number per above), which uniquely identifies a test point for a specific piece of equipment.
- When a new Test Point is created, it requires an equipment reference by logical keys and the test point identifier (TestPointID).
- Arbitrary field values may be specified using the Values parameter, passing one or more field values as XML in a manner detailed below.

- The API web service ensures that the user account has the appropriate security permissions to perform the associated operation, (i.e., Open CP Testing, Add or Clone CP Test Point, Update CP Test Point, and Delete CP Test Point).

Behavior Notes

- The API web service will correct the casing of any values that already exist in the Visions database (i.e., TestPointID, EquipNum, PlantNum, etc.) unless doing so would introduce ambiguity (i.e., conflicting records differing only in case exist), in which case it reads the values case-sensitively.
- The record creation API calls create a record in the associated parent table (CP_TPDetail), as well as zero or more child records in child tables (i.e, CP_TPDate, CP_TPNotes, CP_TPRead, etc.) needed to store values specified in the Values XML string.

List Parameters and Data Format

FieldValues: allow multiple list queries; each of which may take up to four fields in the form of a field name and list of permitted values

Note: All records matching any of the given values in each field are returned.

The list queries return two-dimensional string arrays which duplicate the grid indexes shown in the Visions Client. Row zero of the returned array has the column headers, and each row thereafter is a data row. By default, the returned columns directly mirror the relevant log in the Visions client, but they may be changed or limited by providing a ColumnSet.

ColumnSet: allows you to format the response dataset using the foreign references and field names to join related tables within a List Query call in an API, as well as assign an alias for your foreign reference(s); for example, to return only the logical keys for equipment, specify this ColumnSet value: “PlantNum,EquipNum,EquipType”

Reference the EquipmentListQuery call example for more information.

Data Format

Rows: this parameter provides the Visions Row Counter numbers that correspond to indexes in array parameters. If Rows is omitted, it will be generated automatically.

Columns: this parameter specifies the identity of the Reading Columns that each position in the ReadingsCSV strings will write to. Columns can either be indicated with an integer number (the Column Counter in the Visions Client), a Reading Label (provided that alone identifies a column unambiguously according to the TP's Column Definitions) or the Column Label concatenated with the Reading Units, separated by a semicolon.

Distance: this parameter specifies the distance value with one row corresponding to each value (per the Call examples below).

CP State: this parameter specifies the state value, either on or off, of the cathodic protection at the time of the reading with one row corresponding to each value (per the Call examples below).

ReadingsCSV: CP Test Point readings are passed into and out of the API as a concatenated string of decimal values delimited by semicolons – this is essentially CSV, but a semicolon is used instead of a comma because some locales use the comma in numbers to indicate a decimal place. Each string corresponds to one row, in the same order as the other row parameters.

Numbers in a slot may be omitted, in which case whatever is currently in that position in Visions' database will not change. $NULL or {null} can also be specified in place of a number, in which case that reading position will be blanked in the database.

Query Parameters: When querying test points, all points matching the specified criteria will be returned, along with the most recent set of readings for each returned Test Point in ReadingsCSV format.

Identifiers for a single Test Point may be explicitly specified to QueryCPTP using the EquipNum, EquipType, PlantNum and TestPointID API parameters, or may be generically specified using FieldName1, FieldValues1 and so forth. Examples below document both usages.

XML Field Values

The API calls accept values for arbitrary fields in a very simple XML format, illustrated below and in the examples section:

```
 <met:Values> [Records]
      [Values]
           [Value field="rfEquipment.EquipNum"]V-1102A[/Value]
           [Value field="rfEquipment.PlantNum"]U-1100 - Feed Dryers[/Value]
           [Value field="rfEquipment.kvEquipType"]Vessel[/Value]
           [Value field="TPID"]XTP-BVK44[/Value]
           [Value field="Interval"]5 months[/Value]
     [/Values]
[/Records]</met:Values>
```

This could be sent to create a new Test Point ID for the specified Plant, Equipment Type and Equipment Number.

The only significant tag is "Value", and the only significant attributes are "field" and "fmt". The field attribute is specified in the Tables and Fields group of References section below. This allows the specification of field values in child tables when a test point is created or updated through the CP API.

All date fields require that a format string be included (in the "fmt" attribute) with their value, so that we are able to interpret the value correctly and pre-emptively avoid any issues with date formatting, time zones, locale settings and so forth. The syntax of the date format string is the .NET standard, and is described here:

http://msdn.microsoft.com/en-us/library/az4se3k1

and here:

http://msdn.microsoft.com/en-us/library/8kb3ddd4

References

Data Model Overview

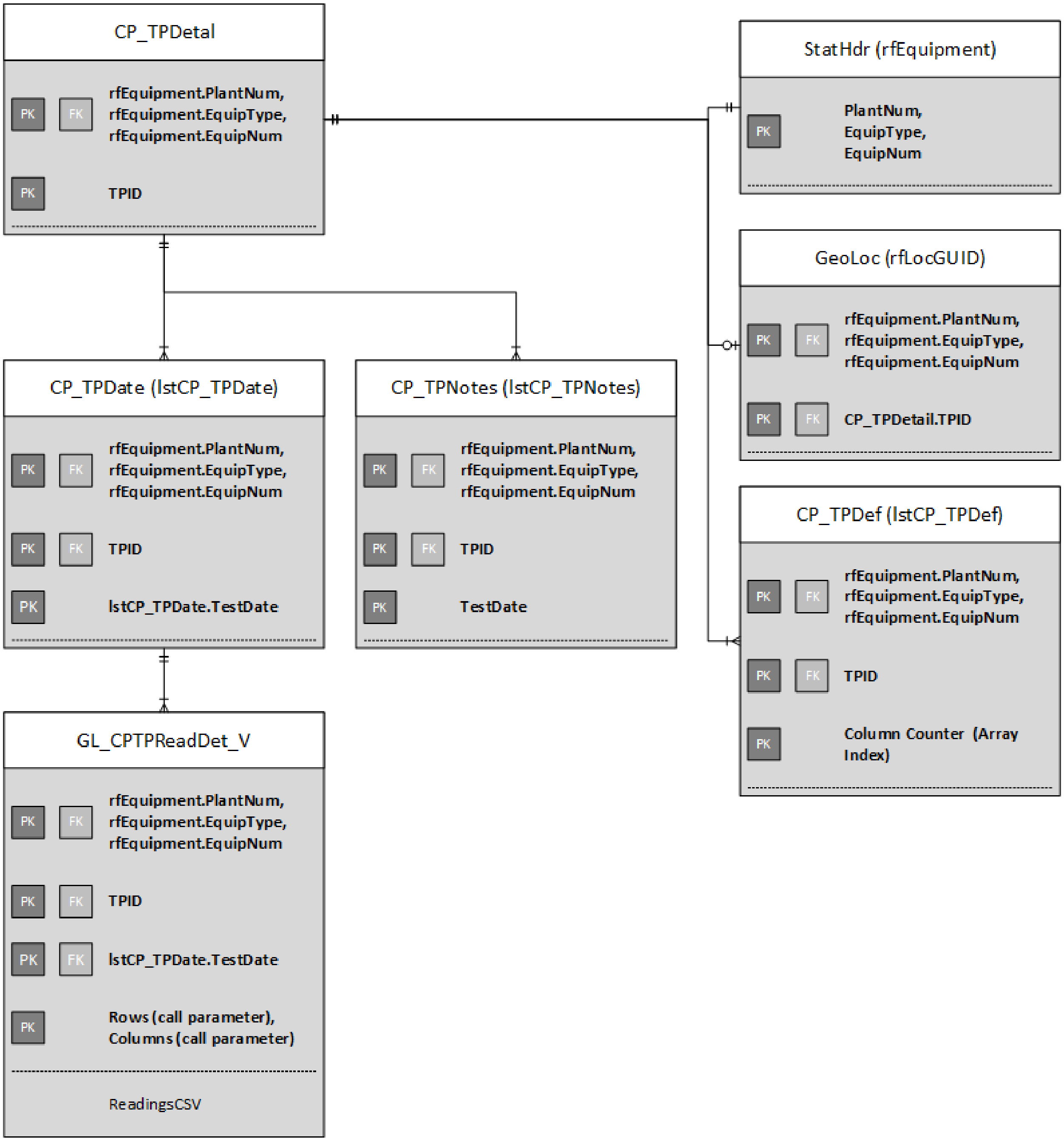

CP_TPDetal
PK FK rfEquipment.PlantNum, rfEquipment.EquipType, rfEquipment.EquipNum
PK TPID
StatHdr (rfEquipment)
PK PlantNum, EquipType, EquipNum
GeoLoc (rfLocGUID)
PK FK rfEquipment.PlantNum, rfEquipment.EquipType, rfEquipment.EquipNum
PK FK CP_TPDetail.TPID
CP_TPDate (lstCP_TPDate)
PK FK rfEquipment.PlantNum, rfEquipment.EquipType, rfEquipment.EquipNum
PK FK TPID
PK lstCP_TPDate.TestDate
CP_TPNotes (lstCP_TPNotes)
PK FK rfEquipment.PlantNum, rfEquipment.EquipType, rfEquipment.EquipNum
PK FK TPID
PK TestDate
CP_TPDef (lstCP_TPDef)
PK FK rfEquipment.PlantNum, rfEquipment.EquipType, rfEquipment.EquipNum
PK FK TPID
PK Column Counter (Array Index)
GL_CPTPReadDet_V
PK FK rfEquipment.PlantNum, rfEquipment.EquipType, rfEquipment.EquipNum
PK FK TPID
PK FK lstCP_TPDate.TestDate
PK Rows (call parameter), Columns (call parameter)
ReadingsCSV

Table and Field Values

The table and field structures and definitions provide the ‘Values’ string within a call. Fields requiring the data to be entered based on a site configured unit of measure are identified by “Unit-Based Field”. Fields requiring the unit of measure value to be entered using the UoM Factors Lookup Data based on the Unit Category are identified by "UoM Factors Unit [Category]".

Limitation: Only the tables and fields listed below are supported for the CP API.

StatHdr (rfEquipment)

Equipment Identification: identify the owning equipment within the TML calls; such as CreateCPTP, CloneCPTP, UpdateCPTP, and DeleteCPTP.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**EquipNum**	("Equipment Number", dtString, Required Field, Read-Only Field, Maximum Length 35)
		Displays the asset or unique identification for the equipment.
string	**EquipType**	("Equipment Type", dtString, Required Field, Read-Only Field, Maximum Length 35, FK (EquipTypeList.EquipType))
		Displays the type of equipment.
string	**PlantNum**	("Plant ID", dtString, Required Field, Read-Only Field, Maximum Length 35, FK (PlantInfo.PlantNum))
		Displays the plant where the equipment is located.

Foreign References		
Table Name	**Call Value**	**Definition**
CP_TPDetail	rfCP_TPDetail	Master table for CP Test Points

CP_TPDetail (rfCP_TPDetail)

CP Testing: the main entity that stores details for each CP test point.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**TPID**	("Test Point ID", dtString, Required Field, Maximum Length 20)
		Enter the test point identification that uniquely identifies each cathodic protection test location for the equipment. **Limitation**: TPID must be unique within each owning equipment.
string	Category	("Category", dtString, Maximum Length 20)
		From a lookup list, select the category of test point.
decimal?	Depth	("Diameter", dtDouble, Unit-Based Field)
		Enter the diameter for the test point.
string	Description	("Description", dtString, Maximum Length 200)
		Enter the descriptive name for the test point.
decimal?	Diameter	("Diameter", dtDouble, Unit-Based Field)
		Enter the diameter for the test point.
string	Drawing No	("Drawing No", dtString, Maximum Length 255)
		Enter the drawing number for the test point.
decimal?	Length	("Length", dtDouble, Unit-Based Field)
		Enter the length for the test point.
string	Location	("Location", dtString, Maximum Length 25)
		Enter the location for the test point.

string	Manufacturer	("Manufacturer", dtString, Maximum Length 50)
		From a lookup list, select the manufacturer of test point.
string	Material	("Material", dtString, Maximum Length 15)
		From a lookup list, select the type of material for the test point.
string	ModelNo	("Model No", dtString, Maximum Length 50)
		Enter the model number for the test point.
DateTime?	NextDate	("Next Date", dtDate)
		Originally set to the date calculated based on the Restricted Interval. This date may be manually changed to match scheduled work.
int?	ReadCount	("Reading Count", dtInteger)
		The number of columns / number of readings to be typically taken for the test point. Note: When setting the number of readings, the definition of the test point must match exactly the reading count and have a label assigned for each column. Reference the CP_Def table for details on the test point definition.
int?	RInterval	("Interval", dtInteger)
		The inspection interval, in months, for the test point.
TG22?	RIntervalUnit	("Interval Unit", dtInteger)
		Select the testing interval for a CP test point or location; choices are: Days, Weeks, Months, Years.
string	TPType	("Test Point Type", dtString, Maximum Length 20)
		From a lookup list, select the type of test point; such as Anode, Junction Box, Test Post, Test Station.
decimal?	Width	("Width", dtDouble, Unit-Based Field)
		Enter the width for the test point.

Foreign Reference		
Table Name	**Call Value**	**Definition**
CP_TPDate	lstCP_TPDate	CP Testing Test Dates
CP_TPDef	lstCP_TPDef	CP Testing Reading (Column) Definitions
CP_TPNotes	lstCP_TPNotes	CP Testing Notes
GeoLoc	rfLocGUID	Geolocation Coordinates

GeoLoc (rfLocGUID)

Geolocation Coordinates: the centralized original GPS and linear coordinates for equipment, work, and trending.

Limitation: The fields listed below are based on the Plant to which the equipment belongs. If the plant is flagged as a Pipeline Plant, then the Linear coordinates are available; otherwise only the original GPS coordinates are available.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)

string	Desc_End	("Description - End", dtString, Maximum Length 100) Enter a description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100) Enter a description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the ending milepost distance (chainage). The lookup list is extracted from the PL_SegMarkers Reference Markers. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20) From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). The lookup list is extracted from the PL_SegMarkers Reference Markers. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field) The ending chainage or linear location. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field) The starting chainage or linear location. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger) From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage. Choices are: Left, Right. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger) From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage. Choices are: Left, Right. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field)

		The original ending GPS Longitude (X) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Longitude (X) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Y_EndOrig	("Latitude - End", dtDouble, Unit-Based Field)
		The original ending GPS Latitude (Y) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field)
		The original starting GPS Latitude (Y) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field)
		The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - End", dtDouble, Unit-Based Field)
		The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign References		
Table Name	**Call Value**	**Definition**
CP_TPDetail	lstCP_TPDetail	Master table for CP Test Points

CP_TPDef (lstCP_TPDef)

CP Test Point Reading Definitions: the definitions for the readings (columns) for test points; represents the Reading Definition grid in the CP Testing detail.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**TPID**	("Test Point ID", dtString, Required Field, Maximum Length 20)
		The test point identifier for the equipment.
decimal?	MaxRead	("Max Read", dtDouble)
		The maximum reading value allowed for the column of the test point. Note: Represented by the MaximumReadings parameter.
decimal?	MinRead	("Min Read", dtDouble)
		The minimum reading value allowed for the column of the test point. Note: Represented by the MinimumReadings parameter.
TG51	ReadLabel	("Reading Label", dtInteger, Required Field)
		Identifies the reading label for the test point. Note: Represented by the ReadingLabels parameter.
string	ReadUnit	("Reading Unit", dtString, Maximum Length 15, UoM Factors Unit [*all*])
		From a lookup list, select the unit of measure for the test point reading, and minimum and maximum reading values within the definition. Note: Represented by the ReadingUnits parameter.

Foreign Reference		
Table Name	**Call Value**	**Definition**
CP_TPDetail	rfCP_TPDetail	Master table for CP Testing; Test Point ID details

CP_TPDate (lstCP_TPDate)

CP Test Point Test Dates: the test dates and associated information for test points; represents the Test Dates grid in the CP Testing detail.

<table>
<tr><th>Data Type</th><th>Field</th><th>Field Information (GUI Reference, GUI Data Type, Size, Purpose)</th></tr>
<tr><td rowspan="2">DateTime</td><td rowspan="2"><b>TestDate</b></td><td>("Test Date", dtDate, Required Field)</td></tr>
<tr><td>Enter the test date for the test point.
<b>Note</b>: This is a unique identifier for each test date within a test point.</td></tr>
<tr><td rowspan="2">string</td><td rowspan="2"><b>TPID</b></td><td>("Test Point ID", dtString, Required Field, Maximum Length 20)</td></tr>
<tr><td>The test point identifier for the equipment.</td></tr>
<tr><td rowspan="2">bool?</td><td rowspan="2">Baseline</td><td>("Baseline", dtInteger, Default Value 0)</td></tr>
<tr><td>Identify whether the reading test date is a baseline measurement; choices are: 0, No, N, 1, Yes, Y.</td></tr>
<tr><td rowspan="2">TG3?</td><td rowspan="2">OnOffFlag</td><td>("Stream", dtInteger, Default Value Internal)</td></tr>
<tr><td>Select the stream that identifies the process stream condition when the test point was measured; choices are:
<table>
<tr><th>Stream</th><th>Definition</th></tr>
<tr><td>Internal</td><td>the test point is located and measured internally within its owning equipment</td></tr>
<tr><td>External (Off)</td><td>the test point is located externally and measured while its owning equipment is off stream</td></tr>
<tr><td>External (On)</td><td>the test point is located externally and measured while its owning equipment is on stream</td></tr>
</table></td></tr>
<tr><td rowspan="2">string</td><td rowspan="2">TestedBy</td><td>("Tested By", dtString, Maximum Length 35)</td></tr>
<tr><td>Enter the name of the person who performed the reading survey for the test point.</td></tr>
</table>

Foreign Reference		
Table Name	**Call Value**	**Definition**
GL_CPTPReadDet_V	lstGL_CPTPReadDet_V	Readings Detail - test dates, distances, CP states and reading coordinates
CP_TPDetail	rfCP_TPDetail	Master table for CP Testing; Test Point ID details

GL_CPTPReadDet_V (lstGL_CPTPReadDet_V)

CP Testing Reading Detail: the individual reading details for test points; represents part of the Readings grid, excluding the reading itself, in the CP Testing detail.

Note: Due to the grid structure of the CP Testing Readings, the readings are managed using a row and column format where the 'Rows' parameter controls the number of rows and the 'Columns' parameter controls in which column the reading will be populated.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)

DateTime	**TestDate**	("Test Date", dtDate, Required Field)
		Enter the test date for the test point. Note: This is a unique identifier for each test date within a test point.
string	**TPID**	("Test Point ID", dtString, Required Field, Maximum Length 20)
		The test point identifier for the equipment.
TG16?	CPState	("CP State", dtInteger)
		Identifies the state of the CP the test point reading; choices are On or Off.
decimal?	Distance	("Distance", dtDouble, Unit-Based Field)
		The distance from the survey point for the test point reading, where a negative number represents upstream.
string	Desc_End	("Description - End", dtString, Maximum Length 100)
		Enter a description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100)
		Enter a description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20)
		From a system lookup list, select milepost marker reference for the ending milepost distance (chainage). The lookup list is extracted from the PL_SegMarkers Reference Markers. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20)
		From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). The lookup list is extracted from the PL_SegMarkers Reference Markers. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field)
		The ending chainage or linear location. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field)
		The starting chainage or linear location. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger)
		From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage. Choices are: Left, Right. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger)

		From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage. Choices are: Left, Right. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field) The distance, perpendicular to the pipe, from where the reference is located. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field) The original ending GPS Longitude (X) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field) The original starting GPS Longitude (X) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Y_EndOrig	("Latitude - End", dtDouble, Unit-Based Field) The original ending GPS Latitude (Y) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field) The original starting GPS Latitude (Y) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field) The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - End", dtDouble, Unit-Based Field) The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign Reference		
Table Name	**Call Value**	**Definition**
CP_TPRead	lstCP_TPRead	Readings - the individual readings for each column and row
CP_TPDate	rfCP_TPDate	Test Dates - the individual test dates for each test point

CP_TPRead (lstCP_TPRead)

CP Testing Readings: the readings for a test point; represents the Reading values in the specified columns of the Readings grid in the CP Testing detail.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
DateTime	**TestDate**	("Test Date", dtDate, Required Field)

		Enter the test date for the test point. **Note**: This is a unique identifier for each test date within a test point.
string	**TPID**	("Test Point ID", dtString, Required Field, Maximum Length 20) The test point identifier for the equipment.
decimal?	Reading	("Reading", dtDouble) The reading for the identified column and row of the test point. Note: Represented by the ReadingsCSV parameter.

Foreign Reference		
Table Name	**Call Value**	**Definition**
GL_CPTPReadDet_V	rfGL_CPTPReadDet_V	Reading Details - the reading distance, state, and coordinates for a test date of a test point.

CP_TPNotes (lstCP_TPNotes)

CP Testing Notes: the individual notes for test points; represents the Notes grid in the CP Testing detail. Dates for test point notes may or may not align with the reading test dates for the test point readings to allow for notes to be added where readings are not surveyed.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
DateTime	**TestDate**	("Test Date", dtDate, Required Field, Read-Only Field) Set the test date for each reading note. **Note**: The test date may different or the same as a reading test date for a test point.
string	**TPID**	("Test Point ID", dtString, Required Field, Maximum Length 20, FK (CP_TPDetail.TPID)) The test point identifier for the equipment.
string	Notes	("Notes", dtString, Maximum Length 4000) Any notes or comments for the notes test date of a test point.
string	NotesCategory	("Notes Category", dtString, Maximum Length 25) From a lookup list, select the category for the notes.

Foreign Reference		
Table Name	**Call Value**	**Definition**
EquipPoint	rfEquipPoint	Master table for TML Trending; TML ID details.

Call Examples

Similar to the Asset API, a debug flag may be included in a call to echo the data in the response message for troubleshooting.

QueryCPTP

Call without Debug Information

```
<!-- This call returns information about CP Test Points in a table
   form, including both their definitions and their latest readings.
   The readings are returned in a CSV string with rows appended
   in sequential order; each row consists of a reading for each
   column described in the "Column_Definitions" entry in the order
   the columns appear there. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:met="http://metegrity.com"
    xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
 <soapenv:Header/>
 <soapenv:Body>
    <met:QueryCPTP>
     <met:SiteName>[Visions SiteName]</met:SiteName>
     <met:UserName>visions_admin</met:UserName>
     <met:Password>mtgy2014</met:Password>
     <met:EquipNum />
     <met:PlantNum>U-1100 - Feed Dryers</met:PlantNum>
     <met:EquipType>Vessel</met:EquipType>
     <met:CPID />
     <met:DateFmt>MM-dd-yy</met:DateFmt>
    </met:QueryCPTP>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- QueryCPTP returns results in an array. The first row gives the column headers for all
    the other rows. The specific columns included can be customized using the ColumnSet
    string. A number of data rows equal to pageSize will be returned, starting at the
    (pageNum x pageSize)th row, and the moreLeft indicator will indicate whether this was
    the final page. QueryCPTP returns (only) the most recent set of readings for each test
    point. They are included as a single column, with the readings in sequential order
    separated by semicolons; as defined in the returned Column_Definitions.  The most
    recent notes for each test point are included as well. Any RTF formatting is stripped
    from the notes content to render them in plain text. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:met="http://metegrity.com"
    xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
 <soapenv:Header/>
 <soapenv:Body>
    <met:QueryCPTP>
     <met:SiteName>[Visions SiteName]</met:SiteName>
     <met:UserName>[Visions UserName]</met:UserName>
     <met:Password>[Visions Password]</met:Password>
     <met:EquipNum />
     <met:PlantNum>Plant 01</met:PlantNum>
     <met:EquipType>Tank</met:EquipType>
     <met:CPID />
     <met:DateFmt>MM-dd-yy</met:DateFmt>
    </met:QueryCPTP>
 </soapenv:Body>
</soapenv:Envelope>

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
```

```
 <s:Body>
<QueryCPTPResponse xmlns="http://metegrity.com">
 <QueryCPTPResult>true</QueryCPTPResult>
 <OutList xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
   xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
   <a:ArrayOfstring>
    <a:string>Plant ID</a:string>
    <a:string>Equipment Type</a:string>
    <a:string>Equipment Number</a:string>
    <a:string>Test Point ID</a:string>
    <a:string>Category</a:string>
    <a:string>Depth</a:string>
    <a:string>Description</a:string>
    <a:string>Diameter</a:string>
    <a:string>Drawing No</a:string>
    <a:string>Last Test Date</a:string>
    <a:string>Last Update</a:string>
    <a:string>Length</a:string>
    <a:string>Location</a:string>
    <a:string>Manufacturer</a:string>
    <a:string>Material</a:string>
    <a:string>Model No</a:string>
    <a:string>Next Date (Calculated)</a:string>
    <a:string>Next Date</a:string>
    <a:string>Reading Count</a:string>
    <a:string>Interval</a:string>
    <a:string>Interval Unit</a:string>
    <a:string>Test Point Type</a:string>
    <a:string>Updated By</a:string>
    <a:string>Width</a:string>
    <a:string>Test Date</a:string>
    <a:string>Baseline</a:string>
    <a:string>Functional Location</a:string>
    <a:string>Stream</a:string>
    <a:string>Orig Due Date</a:string>
    <a:string>Tested By</a:string>
    <a:string>CP State</a:string>
    <a:string>Distance</a:string>
    <a:string>ReadingsCSV</a:string>
    <a:string>Column_Definitions</a:string>
    <a:string>Most Recent Notes Test Date</a:string>
    <a:string>Notes Category</a:string>
    <a:string>Notes</a:string>
   </a:ArrayOfstring>
   <a:ArrayOfstring>
    <a:string>Plant 1</a:string>
    <a:string>Tank</a:string>
    <a:string>TS-61200</a:string>
    <a:string>TS-61200-005</a:string>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string>Measurement of galvanic anode on transition spool</a:string>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string>17/07/2015</a:string>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string>South of 61-TK-200 berm</a:string>
    <a:string>Cott Manufacturing Company</a:string>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string>16/07/2016</a:string>
    <a:string>16/07/2016</a:string>
    <a:string>2</a:string>
    <a:string>12</a:string>
    <a:string>Month</a:string>
    <a:string>Junction Box</a:string>
    <a:string>JDUNCAN</a:string>
    <a:string i:nil="true"/>
    <a:string>30/06/2016</a:string>
    <a:string>False</a:string>
    <a:string i:nil="true"/>
    <a:string>Internal</a:string>
    <a:string i:nil="true"/>
    <a:string>ABC CP Company</a:string>
    <a:string>On</a:string>
    <a:string>1</a:string>
    <a:string>-885</a:string>
    <a:string>Potential-mV</a:string>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
    <a:string i:nil="true"/>
   </a:ArrayOfstring>
 </OutList>
 <moreLeft>false</moreLeft>
 <ErrorMsg>Operation successful.</ErrorMsg>
</QueryCPTPResponse>
 </s:Body>
</s:Envelope>
```

CreateCPTP

Call without Debug Information

```
<!-- The CreateCPTP call is used to create a CP Test Point. This example illustrates
    doing so using explicit parameters to the function, including arrays for each column
    to create the Reading (Column) Definition for the test point. Arrays are parallel, so
    ReadingLabels[2]applies to the same column that MaximumReadings[2]does. Specifying
    arrays with different lengths will result in an error. The LabelColumns parameter
    gives the Visions Column Counters for each CP Column, and will be populated
    implicitly if it is omitted - the user may do this if they don't want to deal with
    the column counters directly. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com"
   xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
 <soapenv:Header/>
 <soapenv:Body>
   <met:CreateCPTP>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>RC-1101</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Rectifier</met:EquipType>
    <met:TestPointID>RC-1101-005</met:TestPointID>
    <met:Interval>12</met:Interval>
    <met:IntervalUnit>Months</met:IntervalUnit>
    <met:LabelColumns>
```

```
      <arr:int>1</arr:int>
      <arr:int>2</arr:int>
    </met:LabelColumns>
    <met:ReadingLabels>
      <arr:string>Coarse Tap</arr:string>
      <arr:string>Fine Tap</arr:string>
    </met:ReadingLabels>
    <met:MinimumReadings>
      <arr:string>1</arr:string>
      <arr:string>1</arr:string>
    </met:MinimumReadings>
    <met:MaximumReadings>
      <arr:string>5</arr:string>
      <arr:string>5</arr:string>
    </met:MaximumReadings>
    <met:ReadingUnits>
      <a:string i:nil="true"/>
      <a:string i:nil="true"/>
    </met:ReadingUnits>
   </met:CreateCPTP>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <CreateCPTPResponse xmlns="http://metegrity.com">
    <CreateCPTPResult>true</CreateCPTPResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </CreateCPTPResponse>
 </s:Body>
</s:Envelope>
```

CreateCPTP (Values XML)

Call without Debug Information

```
<!-- This version of CreateCPTP uses the Values XML to create two CP Test Points in a single
    call, specifying their values as part of the XML. The other parameters - excluding
    connection credentials - thus are not used. Array indices (i.e. "lstCP_TPDef[2]") are
   used to specify multiple rows of information - in this case, the Reading Definitions
    for the CPTP - inside a single block of information, replacing the SOAP arrays in the
    previous call. As such, developers may use this format of specifying information if
    they cannot make SOAP arrays function correctly on their client platform, or if they
    want to create more than one CPTP at once.  GeoLocation information can be specified
    for a CPTP using this expanded format, but not using the simplified parameter format
    above. In this example, only a Longitude value is provided. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com"
   xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
 <soapenv:Header/>
 <soapenv:Body>
   <met:CreateCPTP>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:Values>
      [Records]
```

```
        [Values]
        [Value field="rfEquipment.EquipNum"]V-1102A[/Value]
        [Value field="rfEquipment.PlantNum"]Plant 1[/Value]
        [Value field="rfEquipment.EquipType"]Vessel[/Value]
        [Value field="TPID"]CP-1102A-010[/Value]
        [Value field="Interval"]5 months[/Value]
        [Value field="rfLocGUID.LongitudeStart"]57[/Value]
        [Value field="lstCP_TPDef[1].ReadLabel"]Anode Diameter[/Value]
        [Value field="lstCP_TPDef[1].ReadUnit"]in[/Value]
        [Value field="lstCP_TPDef[1].MinRead"]7[/Value]
        [Value field="lstCP_TPDef[1].MaxRead"]17[/Value]
        [Value field="lstCP_TPDef[2].ReadLabel"]Anode Width[/Value]
        [Value field="lstCP_TPDef[2].ReadUnit"]in[/Value]
        [Value field="lstCP_TPDef[2].MinRead"]8[/Value]
        [Value field="lstCP_TPDef[2].MaxRead"]18[/Value]
        [Value field="lstCP_TPDef[3].ReadLabel"]Anode Depth[/Value]
        [Value field="lstCP_TPDef[3].ReadUnit"]in[/Value]
        [Value field="lstCP_TPDef[3].MinRead"]9[/Value]
        [Value field="lstCP_TPDef[3].MaxRead"]19[/Value]
        [Value field="lstCP_TPDef[4].ReadLabel"]Fine Tap[/Value]
        [Value field="lstCP_TPDef[4].MinRead"]10[/Value]
        [Value field="lstCP_TPDef[4].MaxRead"]20[/Value]
        [/Values]
        [Values]
        [Value field="rfEquipment.EquipNum"]V-1102A[/Value]
        [Value field="rfEquipment.PlantNum"]Plant 1[/Value]
        [Value field="rfEquipment.kvEquipType"]Vessel[/Value]
        [Value field="TPID"]CP-1102A-015[/Value]
        [Value field="Interval"]12 months[/Value]
        [Value field="lstCP_TPDef[1].ReadLabel"]Potential[/Value]
        [Value field="lstCP_TPDef[1].ReadUnit"]V[/Value]
        [Value field="lstCP_TPDef[1].MinRead"]-1100[/Value]
        [Value field="lstCP_TPDef[1].MaxRead"]-850[/Value]
        [Value field="lstCP_TPDef[2].ReadLabel"]Current[/Value]
        [Value field="lstCP_TPDef[2].ReadUnit"]mA[/Value]
        [Value field="lstCP_TPDef[2].MinRead"]150[/Value]
        [Value field="lstCP_TPDef[2].MaxRead"]500[/Value]
        [Value field="lstCP_TPDef[3].ReadLabel"]Voltage[/Value]
        [Value field="lstCP_TPDef[3].ReadUnit"]mV[/Value]
        [Value field="lstCP_TPDef[3].MinRead"]125[/Value]
        [Value field="lstCP_TPDef[3].MaxRead"]200[/Value]
        [/Values]
      [/Records]</met:Values>
   </met:CreateCPTP>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <CreateCPTPResponse xmlns="http://metegrity.com">
    <CreateCPTPResult>true</CreateCPTPResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </CreateCPTPResponse>
 </s:Body>
</s:Envelope>
```

CloneCPTP

Call without Debug Information

```
<!-- This example calls CloneCPTP to create an exact duplicate of an existing test point,
    including the Column Definitions. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:CloneCPTP>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-1102A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:TestPointID>V-1102A-020</met:TestPointID>
    <met:srcEquipNum>V-1102A</met:srcEquipNum>
    <met:srcPlantNum>Plant 1</met:srcPlantNum>
    <met:srcEquipType>Vessel</met:srcEquipType>
    <met:srcTestPointID>V-1102A-030</met:srcTestPointID>
   </met:CloneCPTP>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <CloneCPTPResponse xmlns="http://metegrity.com">
    <CloneCPTPResult>true</CloneCPTPResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </CloneCPTPResponse>
 </s:Body>
</s:Envelope>
```

UpdateCPTP

Call without Debug Information

```
<!-- UpdateCPTP works identically to CreateCPTP, but only updates existing test points
    rather than creating new ones. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com"
   xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
 <soapenv:Header/>
 <soapenv:Body>
   <met:UpdateCPTP>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-1102A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:TestPointID>V-1102A-025</met:TestPointID>
    <met:Interval>5 months</met:Interval>
    <met:IntervalUnit />
    <met:LabelColumns>
       <arr:int>1</arr:int>
       <arr:int>2</arr:int>
       <arr:int>3</arr:int>
    </met:LabelColumns>
    <met:ReadingLabels>
       <arr:string>Anode Length</arr:string>
       <arr:string>Anode Width</arr:string>
       <arr:string>Anode Depth</arr:string>
    </met:ReadingLabels>
    <met:MinimumReadings>
       <arr:string>24</arr:string>
       <arr:string>4</arr:string>
       <arr:string>4</arr:string>
    </met:MinimumReadings>
    <met:MaximumReadings>
       <arr:string>48</arr:string>
       <arr:string>6</arr:string>
       <arr:string>12</arr:string>
    </met:MaximumReadings>
    <met:ReadingUnits>
       <arr:string>in</arr:string>
       <arr:string>in</arr:string>
       <arr:string>in</arr:string>
    </met:ReadingUnits>
   </met:UpdateCPTP>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <UpdateCPTPResponse xmlns="http://metegrity.com">
    <UpdateCPTPResult>true</UpdateCPTPResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </UpdateCPTPResponse>
 </s:Body>
</s:Envelope>
```

DeleteCPTP

Call without Debug Information

```
<!-- DeleteCPTP deletes the test point described by its
   parameters. Multiple test points can be deleted in
   one call using Values syntax if desired. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:DeleteCPTP>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-1102A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:TestPointID>V-1102A-020</met:TestPointID>
   </met:DeleteCPTP>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <DeleteCPTPResponse xmlns="http://metegrity.com">
    <DeleteCPTPResult>true</DeleteCPTPResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </DeleteCPTPResponse>
 </s:Body>
</s:Envelope>
```

AddCPTPReadings

Call without Debug Information

```
<!-- New Readings can be added to an existing Test Point for a new Test Date using the
    AddCPTPReadings API call. Readings can be specified using either parameter syntax or
    Values XML syntax.  The Rows parameter gives the Visions Row Counter numbers that
    correspond to indexes in array parameters. If Rows is omitted, it will be generated
    automatically.  The Columns parameter specifies the identity of the Reading Columns
    that each position in the ReadingsCSV strings will write to. Columns can either be
    indicated with an integer number (the Column Counter in the Visions Client), a Reading
    Label (provided that alone identifies a column unambiguously according to the TP's
    Column Definitions) or the Column Label concatenated with the Reading Units, separated
    by a semicolon. It is the final syntax that is used below.  Distance and CPState give
    those values for each row of readings as a whole. In the example below, there are three
    rows and thus three separate Distance values - one that corresponds to each row.  The
    ReadingsCSV parameter takes an array of strings. Each string corresponds to one row,
    in the same order as the other row parameters. Each row-string must contain a number
    of readings equal to the number of Columns specified, in the order described by the
    Columns array parameter. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com"
   xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
 <soapenv:Header/>
 <soapenv:Body>
   <met:AddCPTPReadings>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-1102A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:TestPointID>V-1102A-015</met:TestPointID>
    <met:TestDate>08/05/2016</met:TestDate>
    <met:Baseline>T</met:Baseline>
    <met:Stream>Internal</met:Stream>
    <met:Inspector>JDuncan Co</met:Inspector>
    <met:Rows>
      <arr:string>1</arr:string>
      <arr:string>2</arr:string>
      <arr:string>3</arr:string>
    </met:Rows>
    <met:Columns>
      <arr:string>Anode Length;in</arr:string>
```

```
       <arr:string>Anode Depth;in</arr:string>
       <arr:string>Anode Width;in</arr:string>
    </met:Columns>
    <met:Distance>
       <arr:string>5</arr:string>
       <arr:string>7</arr:string>
       <arr:string>9</arr:string>
    </met:Distance>
    <met:CPState>
       <arr:string>off</arr:string>
       <arr:string>off</arr:string>
       <arr:string>Off</arr:string>
    </met:CPState>
    <met:ReadingsCSV>
       <arr:string>5.0;5.0;10.0; </arr:string>
       <arr:string>24.0;6.0;12.0</arr:string>
       <arr:string>24.0;6.0;18.0</arr:string>
    </met:ReadingsCSV>
    <met:dateFmt>MM/dd/yyyy</met:dateFmt>
   </met:AddCPTPReadings>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The response indicates the operation succeeded, and gives warning messages whenever
    the readings are outside the Min/Max range for the test point Reading Definition. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <AddCPTPReadingsResponse xmlns="http://metegrity.com">
    <AddCPTPReadingsResult>true</AddCPTPReadingsResult>
    <ErrorMsg>Warning: Reading 5.0 (TPID V-1102A-015, Date 05/08/2016, Row 1, Col 1)
   outside provided range 24-48.
   Operation successful.</ErrorMsg>
   </AddCPTPReadingsResponse>
 </s:Body>
</s:Envelope>
```

AddCPTPReadings (Values XML)

Call without Debug Information

```
<!-- This example AddCPTPReadings call uses array indexing in field names to add three test
    dates' worth of readings to two test points at the same time, as part of one operation.
    Much like with CreateCPTP, the Values syntax can be used to populate GeoLocation
    coordinate fields and other GeoLocation values that aren't included in the explicit
    parameters for ease-of-use. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com"
   xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
 <soapenv:Header/>
 <soapenv:Body>
   <met:AddCPTPReadings>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:dateFmt>MM/dd/yyyy</met:dateFmt>
    <met:Values>[Records]
        [Values]
        [Value field="rfEquipment.EquipNum"]V-1102A[/Value]
        [Value field="rfEquipment.PlantNum"]Plant 1[/Value]
        [Value field="rfEquipment.kvEquipType"]Vessel[/Value]
        [Value field="TPID"]V-1102A-015 [/Value]
        [Value field="lstCP_TPDate[1].TestDate"]09/11/2015[/Value]
        [Value field="lstCP_TPDate[1].Baseline"]T[/Value]
        [Value field="lstCP_TPDate[1].Stream"]Internal[/Value]
        [Value field="lstCP_TPDate[1].Inspector"]Julian[/Value]
        [Value field="lstCP_TPDate[1].Columns[1]"]Anode Diameter;in[/Value]
        [Value field="lstCP_TPDate[1].Columns[2]"]Anode Depth;in[/Value]
        [Value field="lstCP_TPDate[1].Columns[3]"]Anode Width;in[/Value]
        [Value field="lstCP_TPDate[1].Rows[1]"]1[/Value]
        [Value field="lstCP_TPDate[1].Rows[2]"]2[/Value]
        [Value field="lstCP_TPDate[1].Rows[3]"]3[/Value]
        [Value field="lstCP_TPDate[1].Distance[1]"]5[/Value]
        [Value field="lstCP_TPDate[1].Distance[2]"]7[/Value]
        [Value field="lstCP_TPDate[1].Distance[3]"]9[/Value]
        [Value field="lstCP_TPDate[1].CPState[1]"]on[/Value]
        [Value field="lstCP_TPDate[1].CPState[2]"]off[/Value]
        [Value field="lstCP_TPDate[1].CPState[3]"]Off[/Value]
        [Value field="lstCP_TPDate[1].ReadingsCSV[1]"]20.0;5.0;12.0[/Value]
        [Value field="lstCP_TPDate[1].ReadingsCSV[2]"]22.0;6.0;12.0[/Value]
        [Value field="lstCP_TPDate[1].ReadingsCSV[3]"]22.0;6.0;10.0[/Value]
        [Value field="lstCP_TPDate[1].LongitudeStart"]77.254[/Value]
        [Value field="lstCP_TPDate[1].LatitudeStart"]63.783[/Value]
        [Value field="lstCP_TPDate[1].ElevationStart"]6342[/Value]
        [Value field="lstCP_TPDate[2].TestDate"]04/04/2016[/Value]
        [Value field="lstCP_TPDate[2].Baseline"]T[/Value]
        [Value field="lstCP_TPDate[2].Stream"]Internal[/Value]
        [Value field="lstCP_TPDate[2].Inspector"]Julian[/Value]
        [Value field="lstCP_TPDate[2].Columns[1]"]Anode Diameter;mA[/Value]
        [Value field="lstCP_TPDate[2].Columns[2]"]Anode Depth;mV[/Value]
        [Value field="lstCP_TPDate[2].Columns[3]"]Anode Width;Amps[/Value]
        [Value field="lstCP_TPDate[2].Distance[1]"]33[/Value]
        [Value field="lstCP_TPDate[2].Distance[2]"]34[/Value]
        [Value field="lstCP_TPDate[2].Distance[3]"]35[/Value]
        [Value field="lstCP_TPDate[2].CPState[1]"]off[/Value]
        [Value field="lstCP_TPDate[2].CPState[2]"]on[/Value]
        [Value field="lstCP_TPDate[2].CPState[3]"]on[/Value]
        [Value field="lstCP_TPDate[2].ReadingsCSV[1]"]18.0;5.0;11.0[/Value]
        [Value field="lstCP_TPDate[2].ReadingsCSV[2]"]21.5;5.5;11.5[/Value]
        [Value field="lstCP_TPDate[2].ReadingsCSV[3]"]21.2;5.0;11.8[/Value]
        [/Values]
      [/Records]</met:Values>
   </met:AddCPTPReadings>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<!-- The response indicates the operation succeeded, and gives warning messages whenever
    the readings are outside the Min/Max range for the test point Reading Definition. -->
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <AddCPTPReadingsResponse xmlns="http://metegrity.com">
    <AddCPTPReadingsResult>true</AddCPTPReadingsResult>
    <ErrorMsg>Warning: Reading 5.0 (TPID V-1102A-015, Date 05/08/2016, Row 1, Col 1)
   outside provided range 24-48.
   Operation successful.</ErrorMsg>
```

```
    </AddCPTPReadingsResponse>
 </s:Body>
</s:Envelope>
```

EditCPTPReadings

Call without Debug Information

```
<!-- EditCPTPReadings works similarly to the Add API, but modifies reading Test Dates that
    already exist in the system. It can be used to edit actual numbers already populated,
    or to add new column values to an already existing Test Date. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:met="http://metegrity.com"
    xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
 <soapenv:Header/>
 <soapenv:Body>
    <met:EditCPTPReadings>
     <met:SiteName>[Visions SiteName]</met:SiteName>
     <met:UserName>[Visions UserName]</met:UserName>
     <met:Password>[Visions Password]</met:Password>
     <met:dateFmt>MM/dd/yyyy</met:dateFmt>
     <met:Values>[Records]
         [Values]
         [Value field="rfEquipment.EquipNum"]V-2102A[/Value]
         [Value field="rfEquipment.PlantNum"]Plant 2[/Value]
         [Value field="rfEquipment.kvEquipType"]Vessel[/Value]
         [Value field="TPID"]V-2102A-025[/Value]
         [Value field="lstCP_TPDate[1].TestDate"]09/11/1997[/Value]
         [Value field="lstCP_TPDate[1].Baseline"]T[/Value]
         [Value field="lstCP_TPDate[1].Stream"]Internal[/Value]
         [Value field="lstCP_TPDate[1].Inspector"]JSmith[/Value]
         [Value field="lstCP_TPDate[1].Columns[1]"]Anode Diameter;mA[/Value]
         [Value field="lstCP_TPDate[1].Columns[2]"]Anode Depth;mV[/Value]
         [Value field="lstCP_TPDate[1].Columns[3]"]Anode Width;Amps[/Value]
         [Value field="lstCP_TPDate[1].Rows[1]"]1[/Value]
         [Value field="lstCP_TPDate[1].Rows[2]"]2[/Value]
         [Value field="lstCP_TPDate[1].Rows[3]"]3[/Value]
         [Value field="lstCP_TPDate[1].Distance[1]"]5[/Value]
         [Value field="lstCP_TPDate[1].Distance[2]"]7[/Value]
         [Value field="lstCP_TPDate[1].Distance[3]"]9[/Value]
         [Value field="lstCP_TPDate[1].CPState[1]"]on[/Value]
         [Value field="lstCP_TPDate[1].CPState[2]"]off[/Value]
         [Value field="lstCP_TPDate[1].CPState[3]"]Off[/Value]
         [Value field="lstCP_TPDate[1].ReadingsCSV[1]"]12.1;9.4;10.6[/Value]
         [Value field="lstCP_TPDate[1].ReadingsCSV[2]"]12.2;9.5;10.7[/Value]
         [Value field="lstCP_TPDate[1].ReadingsCSV[3]"]12.3;9.6;10.8[/Value]
         [Value field="lstCP_TPDate[1].LongitudeStart[1]"]77[/Value]
         [Value field="lstCP_TPDate[1].LatitudeStart[1]"]45[/Value]
         [Value field="lstCP_TPDate[1].ElevationStart[1]"]4571[/Value]
         [Value field="lstCP_TPDate[1].CenterlineFlag[1]"]F[/Value]
         [Value field="lstCP_TPDate[1].OffsetDir_Start[1]"]Left[/Value]
         [Value field="lstCP_TPDate[1].OffsetDist_Start[1]"]90[/Value]
         [Value field="lstCP_TPDate[1].LongitudeStart[2]"]78[/Value]
         [Value field="lstCP_TPDate[1].LatitudeStart[2]"]47[/Value]
         [Value field="lstCP_TPDate[1].ElevationStart[2]"]4572[/Value]
         [Value field="lstCP_TPDate[1].CenterlineFlag[2]"]F[/Value]
         [Value field="lstCP_TPDate[1].OffsetDir_Start[2]"]Left[/Value]
         [Value field="lstCP_TPDate[1].OffsetDist_Start[2]"]91[/Value]
         [Value field="lstCP_TPDate[1].LongitudeStart[3]"]79[/Value]
```

```
        [Value field="lstCP_TPDate[1].LatitudeStart[3]"]49[/Value]
        [Value field="lstCP_TPDate[1].ElevationStart[3]"]4573[/Value]
        [Value field="lstCP_TPDate[1].CenterlineFlag[3]"]F[/Value]
        [Value field="lstCP_TPDate[1].OffsetDir_Start[3]"]Left[/Value]
        [Value field="lstCP_TPDate[1].OffsetDist_Start[3]"]92[/Value]
        [Value field="lstCP_TPDate[2].TestDate"]04/04/2005[/Value]
        [Value field="lstCP_TPDate[2].Baseline"]T[/Value]
        [Value field="lstCP_TPDate[2].Stream"]Internal[/Value]
        [Value field="lstCP_TPDate[2].Inspector"]Julian[/Value]
        [Value field="lstCP_TPDate[2].Columns[1]"]Anode Diameter;mA[/Value]
        [Value field="lstCP_TPDate[2].Columns[2]"]Anode Depth;mV[/Value]
        [Value field="lstCP_TPDate[2].Columns[3]"]Anode Width;Amps[/Value]
        [Value field="lstCP_TPDate[2].Columns[4]"]Fine Tap;Volts[/Value]
        [Value field="lstCP_TPDate[2].Distance[1]"]33[/Value]
        [Value field="lstCP_TPDate[2].Distance[2]"]34[/Value]
        [Value field="lstCP_TPDate[2].Distance[3]"]35[/Value]
        [Value field="lstCP_TPDate[2].CPState[1]"]off[/Value]
        [Value field="lstCP_TPDate[2].CPState[2]"]on[/Value]
        [Value field="lstCP_TPDate[2].CPState[3]"]on[/Value]
        [Value field="lstCP_TPDate[2].ReadingsCSV[1]"]9.1;9.2;9.3;12.7[/Value]
        [Value field="lstCP_TPDate[2].ReadingsCSV[2]"]10.1;10.2;10.3;12.8[/Value]
        [Value field="lstCP_TPDate[2].ReadingsCSV[3]"]11.5;11.6;$NULL;12.9[/Value]
        [/Values]
        [Values]
        [Value field="rfEquipment.EquipNum"]V-2102A[/Value]
        [Value field="rfEquipment.PlantNum"]Plant 2[/Value]
        [Value field="rfEquipment.kvEquipType"]Vessel[/Value]
        [Value field="TPID"]V-2102A-025[/Value]
        [Value field="lstCP_TPDate[1].TestDate"]11/11/2001[/Value]
        [Value field="lstCP_TPDate[1].Baseline"]F[/Value]
        [Value field="lstCP_TPDate[1].Stream"]ExternalOff[/Value]
        [Value field="lstCP_TPDate[1].Inspector"]Julian[/Value]
        [Value field="lstCP_TPDate[1].Columns[1]"]Coarse Tap [/Value]
        [Value field="lstCP_TPDate[1].Columns[2]"]Current;mV[/Value]
        [Value field="lstCP_TPDate[1].Columns[3]"]Potential;V[/Value]
        [Value field="lstCP_TPDate[1].Rows[1]"]1[/Value]
        [Value field="lstCP_TPDate[1].Rows[2]"]2[/Value]
        [Value field="lstCP_TPDate[1].Rows[3]"]3[/Value]
        [Value field="lstCP_TPDate[1].Distance[1]"]5[/Value]
        [Value field="lstCP_TPDate[1].Distance[2]"]7[/Value]
        [Value field="lstCP_TPDate[1].Distance[3]"]9[/Value]
        [Value field="lstCP_TPDate[1].CPState[1]"]on[/Value]
        [Value field="lstCP_TPDate[1].CPState[2]"]off[/Value]
        [Value field="lstCP_TPDate[1].CPState[3]"]Off[/Value]
        [Value field="lstCP_TPDate[1].ReadingsCSV[1]"]12.2;13.3;14.4[/Value]
        [Value field="lstCP_TPDate[1].ReadingsCSV[2]"]12.5;13.6;14.7[/Value]
        [Value field="lstCP_TPDate[1].ReadingsCSV[3]"]12.8;13.9;14.0[/Value]
        [/Values]
      [/Records]</met:Values>
   </met:EditCPTPReadings>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <EditCPTPReadingsResponse xmlns="http://metegrity.com">
    <EditCPTPReadingsResult>true</EditCPTPReadingsResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </EditCPTPReadingsResponse>
 </s:Body>
</s:Envelope>
```

DeleteCPTPReadings

Call without Debug Information

```
<!-- The DeleteCPTPReadings call removes a Test Date and all associated readings from a
    Test Point. By specifying the optional Rows parameter, specific rows can be removed
    from a Test Date without removing the Test Date overall. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com"
   xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
 <soapenv:Header/>
 <soapenv:Body>
   <met:DeleteCPTPReadings>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-1102A</met:EquipNum>
    <met:PlantNum>Plant 2</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:TestPointID>V-2102A-025</met:TestPointID>
    <met:TestDate>11/11/2001</met:TestDate>
    <!--
    <met:Rows>
       <arr:string>?</arr:string>
    </met:Rows> -->
    <met:dateFmt>MM/dd/yyyy</met:dateFmt>
   </met:DeleteCPTPReadings>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <DeleteCPTPReadingsResponse xmlns="http://metegrity.com">
    <DeleteCPTPReadingsResult>true</DeleteCPTPReadingsResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </DeleteCPTPReadingsResponse>
 </s:Body>
</s:Envelope>
```

AddCPTPNotes

Call without Debug Information

```
<!-- AddCPTPNotes text allows annotating text to be added to a specific Test Date. A Test
    Date can have either readings or notes, or both. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:AddCPTPNotes>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-1102A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:TestPointID>V-1102A-025</met:TestPointID>
    <met:TestDate>08/05/2014</met:TestDate>
    <met:dateFmt>MM/dd/yyyy</met:dateFmt>
    <met:Category>Access</met:Category>
    <met:Notes>Unable to access reference test post due to flooding.</met:Notes>
   </met:AddCPTPNotes>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <AddCPTPNotesResponse xmlns="http://metegrity.com">
    <AddCPTPNotesResult>true</AddCPTPNotesResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </AddCPTPNotesResponse>
 </s:Body>
</s:Envelope>
```

Additional Information

Visions Enterprise Help File

Lookup Data > F3 Lookup Lists

Cathodic Protection

Cathodic Protection > CP Testing

General Trending API

Overview

Several application programming interfaces (API) are available to allow you to perform certain actions without using the Visions Enterprise user interface (UI). This allows third party developers to use web service calls to get or update Visions data.

The Visions API for General Trending allows you to manage trend points; such as creating, updating, or querying trend point identifiers, readings, and trend types data.

Access

Navigation

The web service calls are expressed as a C# interface, with the individual calls, conventions, references, and examples documented below.

```
public interface IVisAPI {
...
}
```

Conditions

1. The General Trending API is a separate module which must be active to use its calls.
2. Some equipment types may not own Trend Points, this includes any types derived from the following supplied types: Circuit, and Major.
3. Security to perform calls are dependent upon the security access for the username passed into a call. For example: if you do not have Delete General Trending Trend Point privileges, you will not be able to call the DeleteGTTP call successfully. Security functions for General Trending are:
    a. **Open General Trending**: controls access to the General Trending, all calls
    b. **Add, Change or Delete General Trending**: controls general trending test point creation, update or deletion
    c. **Add or Change Gen Trending Type**: controls creation or modification to the general trending trend types
    d. **Delete Gen Trending Type**: controls deletion of general trending trend types

Input

The General Trending API provides the following calls:

- **QueryGTTrendType**: query the General Trending Trend Types, returning the value(s) in a table form

- **CreateGTTrendType**: create the identified General Trending trend type(s); simulates the ‘New’ action on the General Trending Trending Types detail

  Security: Permission is dependent upon the ‘Add or Change Gen Trending Type’ function.

- **UpdateGTTrendType**: update an existing General Trending trend type

  Security: Permission is dependent upon the ‘Add or Change Gen Trending Type’ function.

- **DeleteGTTrendType**: deletes the identified General Trending trend type(s); simulates the ‘Delete’ action on the General Trending Trending Types detail

  Security: Permission is dependent upon the ‘Delete Gen Trending Type’ function.

- **QueryGTTP**: query the General Trending trend points, returning the value(s) in a table form, including both their definitions and their latest readings
- **CreateGTTP**: create the identified General Trending trend point(s); simulates the ‘New’ action on the General Trending Log.

  Security: Permission is dependent upon the ‘Add, Change or Delete General Trending’ function.

- **UpdateGTTP**: update an existing General Trending trend point; excludes changes to readings and owning equipment

  Security: Permission is dependent upon the ‘Add, Change or Delete General Trending’ function.

- **DeleteGTTP**: deletes the identified General Trending trend point(s); simulates the ‘Delete’ action on the General Trending Log

  Security: Permission is dependent upon the ‘Add, Change or Delete General Trending’ function.

- **AddGTTPReadings**: add new readings to a test date for General Trending trend point(s); simulates the addition of ‘Readings’ on the General Trending detail
- **EditGTTPReadings**: edit readings for the General Trending trend point(s); either edit existing readings or add new row values to an existing test date
- **DeleteGTTPReadings**: delete a test date and all associated readings from General Trending trend point(s); optionally delete specific rows of readings without deleting the full test date

Calls

```
bool QueryGTTP(string SiteName, string UserName, string Password, string EquipNum, string
PlantNum, string EquipType, string GTTP, string FieldName1, string[] FieldValues1, string
FieldName2, string[] FieldValues2, string FieldName3, string[] FieldValues3, string
FieldName4, string[] FieldValues4, string DateFmt, string ColumnSet, int pageSize, int
pageNum, out string[][] OutList, out bool moreLeft, out string ErrorMsg, string MapID);

bool QueryGTTrendType(string SiteName, string UserName, string Password, out string[]
TrendTypes, out string[] TypeDescriptions, out string ErrorMsg);

bool CreateGTTrendType(string SiteName, string UserName, string Password,  string
TrendType, string TypeDescription,  string Values, string MapID, out string ErrorMsg);

bool UpdateGTTrendType(string SiteName, string UserName, string Password,  string
TrendType, string newTypeName, string TypeDescription,  string Values, string MapID, out
string ErrorMsg);
```

```
bool DeleteGTTrendType(string SiteName, string UserName, string Password, string
TrendType, string Values, string MapID, out string ErrorMsg);

bool CreateGTTP(string SiteName, string UserName, string Password, string EquipNum,
string PlantNum, string EquipType, string GTTP, string TrendType, string Description,
string MinRead, string MaxRead, string ReadingUnit, string NextTestDateDue, string
RInterval, string RIntervalUnit, string dateFmt, string Values, out string ErrorMsg,
string MapID);

bool UpdateGTTP(string SiteName, string UserName, string Password, string EquipNum,
string PlantNum, string EquipType, string GTTP, string TrendType, string Description,
string MinRead, string MaxRead, string ReadingUnit, string NextTestDateDue, string
RInterval, string RIntervalUnit, string dateFmt, string Values, out string ErrorMsg,
string MapID);

bool DeleteGTTP(string SiteName, string UserName, string Password, string EquipNum,
string PlantNum, string EquipType, string GTTP, string Values, out string ErrorMsg,
string MapID);

bool AddGTTPReadings(string SiteName, string UserName, string Password, string EquipNum,
string PlantNum, string EquipType, string GTTP, string TestDate, string Baseline, string
Stream, string TestedBy, string Comments, string ReadingsCSV, string ReadTypesCSV, string
Values, string dateFmt, out string ErrorMsg, string MapID);

bool EditGTTPReadings(string SiteName, string UserName, string Password, string EquipNum,
string PlantNum, string EquipType, string GTTP, string TestDate, string Baseline, string
Stream, string TestedBy, string Comments, string ReadingsCSV, string ReadTypesCSV, string
Values, string dateFmt, out string ErrorMsg, string MapID);

bool DeleteGTTPReadings(string SiteName, string UserName, string Password, string
EquipNum, string PlantNum, string EquipType, string GTTP, string TestDate, string
dateFmt, string Values, out string ErrorMsg, string MapID);
```

Calling Conventions

The web service is a standard SOAP web service and may be consumed by anything that can consume SOAP web services.

The following conventions are specific to the General Trending (GT) API:

- API calls which reference a piece of equipment do so by its three (3) logical keys – PlantNum, EquipType, and EquipNum – not our internal Visions account identifier.
- API calls to interact with existing GT Trend Points must reference them by the logical key of GTTP (Trend Point) and asset identification (Plant, Equipment Type, and Equipment Number per above), which uniquely identifies a test point for a specific piece of equipment.
- When a new Trend Point is created, it requires an equipment reference by logical keys, the test point identifier (GTTP), and the trend type (TrendType).
- Arbitrary field values may be specified using the Values parameter, passing one or more field values as XML in a manner detailed below.

- The API web service ensures that the user account has the appropriate security permissions to perform the associated operation, (i.e., Open General Trending, Add, Change or Delete General Trending, Add or Change Gen Trending Type, and Delete Gen Trending Type).

Behavior Notes

- The API web service will correct the casing of any values that already exist in the Visions database (i.e., GTTP, TrendType, EquipNum, PlantNum, etc.) unless doing so would introduce ambiguity (i.e., conflicting records differing only in case exist), in which case it reads the values case-sensitively.
- The record creation API calls creates a record in the associated parent table (TrendTestPoint), as well as zero or more child records in child tables (i.e, TrendDates, TrendRead, GeoLoc, etc.) needed to store values specified in the Values XML string.

List Parameters and Data Format

FieldValues: allow multiple list queries; each of which may take up to four fields in the form of a field name and list of permitted values

Note: All records matching any of the given values in each field are returned.

The list queries return two-dimensional string arrays which duplicate the grid indexes shown in the Visions Client. Row zero of the returned array has the column headers, and each row thereafter is a data row. By default, the returned columns directly mirror the relevant log in the Visions client, but they may be changed or limited by providing a ColumnSet.

ColumnSet: allows you to format the response dataset using the foreign references and field names to join related tables within a List Query call in an API, as well as assign an alias for your foreign reference(s); for example, to return only the logical keys for equipment, specify this ColumnSet value: “PlantNum,EquipNum,EquipType”

Reference the EquipmentListQuery call example for more information.

Data Format

Rows: this parameter provides the Visions Row Counter numbers that correspond to indexes in array parameters. If Rows is omitted, it will be generated automatically.

Columns: this parameter specifies the identity of the Reading Columns that each position in the ReadingsCSV strings will write to. Columns can either be indicated with an integer number (the Column Counter in the Visions Client), a Reading Label (provided that alone identifies a column unambiguously according to the TP's Column Definitions) or the Column Label concatenated with the Reading Units, separated by a semicolon.

ReadTypesCSV: General Trending Trend Point readings are passed into and out of the API as a concatenated string of decimal values delimited by semicolons – this is essentially CSV, but a semicolon is used instead of a comma because some locales use the comma in numbers to indicate a decimal place. Each string corresponds to one row, in the same order as the other row parameters.

ReadingsCSV: General Trending Trend Point readings are passed into and out of the API as a concatenated string of decimal values delimited by semicolons – this is essentially CSV, but a semicolon is used instead of a comma because some locales use the comma in numbers to indicate a decimal place. Each string corresponds to one row, in the same order as the other row parameters.

Numbers in a slot may be omitted, in which case whatever is currently in that position in Visions' database will not change. $NULL or {null} can also be specified in place of a number, in which case that reading position will be blanked in the database.

Query Parameters: When querying trend points, all points matching the specified criteria will be returned, along with the most recent set of readings for each returned Test Point in ReadingsCSV format.

Identifiers for a single Trend Point may be explicitly specified to QueryGTTP using the EquipNum, EquipType, PlantNum and GTTP (Trend Point) API parameters, or may be generically specified using FieldName1, FieldValues1 and so forth. Examples below document both usages.

XML Field Values

The API calls accept values for arbitrary fields in a very simple XML format, illustrated below and in the examples section:

```
 <met:Values> [Records]
    [Values]
        [Value field="rfEquipment.EquipNum"]P-1102A[/Value]
        [Value field="rfEquipment.PlantNum"]Plant 1[/Value]
        [Value field="rfEquipment.kvEquipType"]Pipe[/Value]
        [Value field="GTTP"]P1102A-005 [/Value]
       [Value field="TrendType"]W/C ratio[/Value]
    [/Values]
[/Records]</met:Values>
```

This could be sent to create a new Trend Point for the specified Plant, Equipment Type and Equipment Number.

The only significant tag is "Value", and the only significant attributes are "field" and "fmt". The field attribute is specified in the Tables and Fields group of References section below. This allows the specification of field values in child tables when a test point is created or updated through the General Trending API.

All date fields require that a format string be included (in the "fmt" attribute) with their value, so that we are able to interpret the value correctly and pre-emptively avoid any issues with date formatting, time zones, locale settings and so forth. The syntax of the date format string is the .NET standard, and is described here:

http://msdn.microsoft.com/en-us/library/az4se3k1

and here:

http://msdn.microsoft.com/en-us/library/8kb3ddd4

References

Data Model Overview

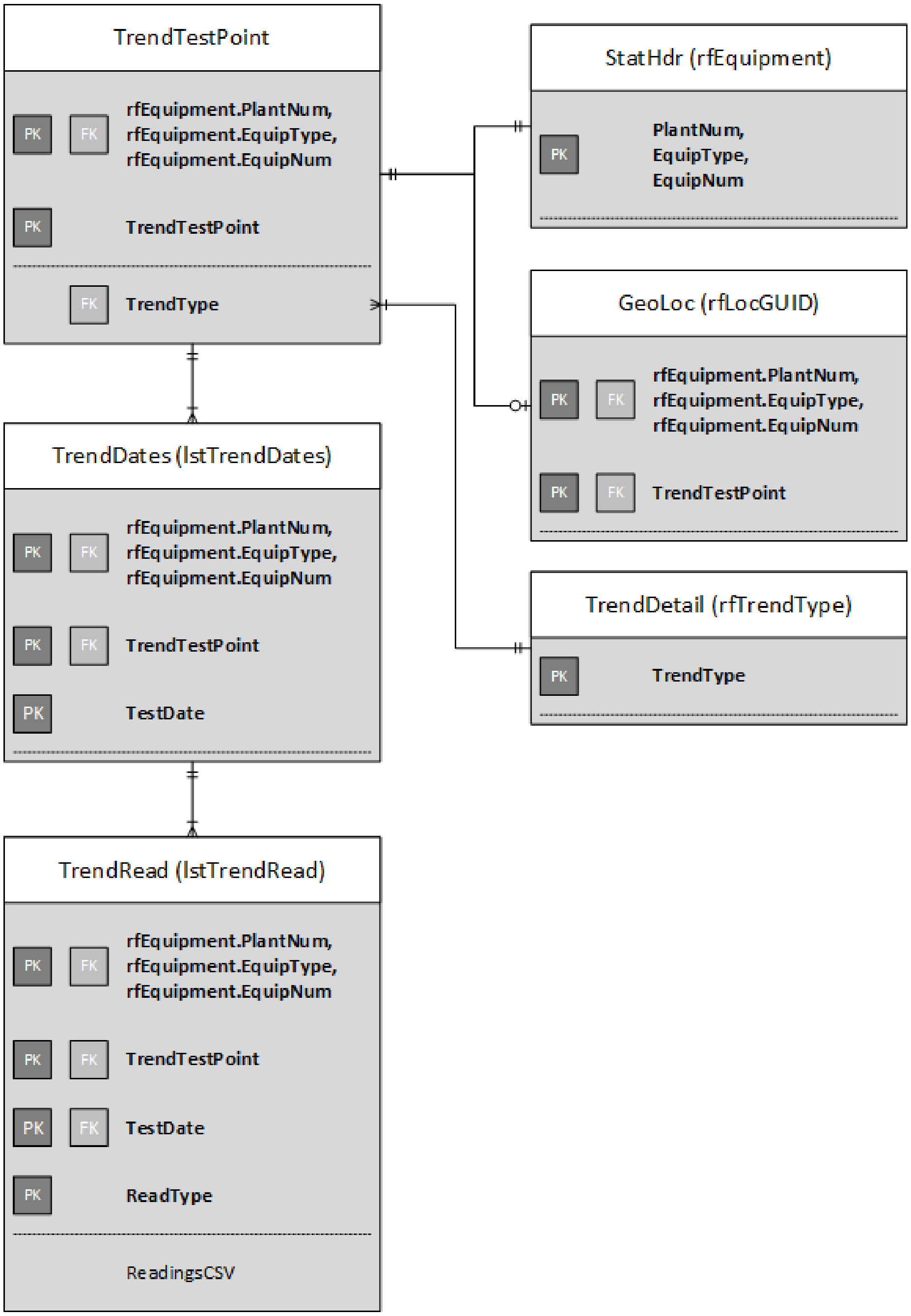


Table and Field Values

The table and field structures and definitions provide the ‘Values’ string within a call. Fields requiring the data to be entered based on a site configured unit of measure are identified by “Unit-Based Field”. Fields requiring the unit of measure value to be entered using the UoM Factors Lookup Data based on the Unit Category are identified by "UoM Factors Unit [Category]".

Limitation: Only the tables and fields listed below are supported for the GT API.

StatHdr (rfEquipment)

Equipment Identification: identify the owning equipment within the General Trending calls; such as CreateGTTP, UpdateGTTP, and DeleteGTTP.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**EquipNum**	("Equipment Number", dtString, Required Field, Read-Only Field, Maximum Length 35)
		Displays the asset or unique identification for the equipment.
string	**EquipType**	("Equipment Type", dtString, Required Field, Read-Only Field, Maximum Length 35, FK (EquipTypeList.EquipType))
		Displays the type of equipment.
string	**PlantNum**	("Plant ID", dtString, Required Field, Read-Only Field, Maximum Length 35, FK (PlantInfo.PlantNum))
		Displays the plant where the equipment is located.

Foreign References		
Table Name	**Call Value**	**Definition**
TrendTestPoint	rfTrendTestPoint	Master table for General Trending Test Points

TrendTestPoint

General Trending: the main entity that stores details for each trend point.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	**TrendTestPoint**	("Trend Test Point", dtString,, Required Field, Maximum Length 15)
		Enter the test point identification that uniquely identifies each general trending location for the equipment. **Limitation**: TrendTestPoint must be unique within each owning equipment.
string	Description	("Description", dtString, Maximum Length 200)
		Enter the descriptive name for the test point.
decimal?	MaxRead	("Maximum Reading", dtDouble, Unit-Based Field)
		The maximum measurement or reading value for each trend point.
DateTime?	NextDate	("Next Test Due Date", dtDate)
		Originally set to the date calculated based on the Restricted Interval. This date may be manually changed to match scheduled work.
string	ReadingUnit	("Reading Unit", dtString, Maximum Length 15, UoM Factors Unit [*all*])
		From a lookup list, select the unit of measurement for the measurement or reading value.
int?	RInterval	("Interval", dtInteger)
		The inspection interval, in months, for the trend point.
TG22?	RIntervalUnit	("Interval Unit", dtInteger)

		Select the testing interval for a trend point; choices are: Days, Weeks, Months, Years.
string	TrendType	("Trend Type", dtString, Required Field, Maximum Length 15, FK (rfTrendType.TrendType)) Select the type of trending for a test point. The available items are from the Trend Type data.

Foreign Reference		
Table Name	**Call Value**	**Definition**
TrendDates	lstTrendDates	General Trending Dates
TrendDetail	rfTrendType	General Trending Types - master list of types of trending
GeoLoc	rfLocGUID	Geolocation Coordinates

TrendDetail (rfTrendType)

General Trending Types: identify the types of trending within the General Trending calls; such as QueryGTTrendTypes, CreateGTTrendTypes, UpdateGTTrendTypes, and DeleteGTTrendTypes.

Data Type	**Field**	**Field Information (GUI Reference, GUI Data Type, Size, Purpose)**
string	**TrendType**	("Trend Type", dtString, Required Field, Maximum Length 15, FK (TrendTestPoint.TrendType)) Select the type of trending for a test point. The available items are from the Trend Type data. Limitation: Must be unique within a site (schema).
string	Description	("Description", dtString, Maximum Length 200) Enter a description for a trending type. Example: Coupon measurement, utility pigging, etc.

Foreign Reference		
Table Name	**Call Value**	**Definition**
TrendDetail	lstTrendType	General Trending - master details for trend points
GeoLoc	rfLocGUID	Geolocation Coordinates

GeoLoc (rfLocGUID)

Geolocation Coordinates: the centralized original GPS and linear coordinates for equipment, work, and trending.

Limitation: The fields listed below are based on the Plant to which the equipment belongs. If the plant is flagged as a Pipeline Plant, then the Linear coordinates are available; otherwise only the original GPS coordinates are available.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
string	Desc_End	("Description - End", dtString, Maximum Length 100)

		Enter a description for the ending reference point of the GPS chainage Example: The southwest corner of the vessel.
string	Desc_Start	("Description - Start", dtString, Maximum Length 100)
		Enter a description for the starting reference point of the GPS coordinate. Example: The northeast corner of the vessel.
string	MarkerID_End	("Reference ID - End", dtString, Maximum Length 20)
		From a system lookup list, select milepost marker reference for the ending milepost distance (chainage). The lookup list is extracted from the PL_SegMarkers Reference Markers. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
string	MarkerID_Start	("Reference ID - Start", dtString, Maximum Length 20)
		From a system lookup list, select milepost marker reference for the starting milepost distance (chainage). The lookup list is extracted from the PL_SegMarkers Reference Markers. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	MP_End	("MP (Chainage) - End", dtDouble, Unit-Based Field)
		The ending chainage or linear location. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	MP_Start	("MP (Chainage) - Start", dtDouble, Unit-Based Field)
		The starting chainage or linear location. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
TG56?	OffsetDir_End	("Offset Direction - End", dtInteger)
		From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage. Choices are: Left, Right. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
TG56?	OffsetDir_Start	("Offset Direction - Start", dtInteger)
		From a system lookup, select the direction from where the reference is located. Perspective is always relative to the increasing direction of chainage. Choices are: Left, Right. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	OffsetDist_End	("Offset Distance - End", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	OffsetDist_Start	("Offset Distance - Start", dtDouble, Unit-Based Field)
		The distance, perpendicular to the pipe, from where the reference is located. Part of the Linear coordinates and only updatable when the PL module is active and the equipment belongs to a Pipeline plant.
decimal?	X_EndOrig	("Longitude - End", dtDouble, Unit-Based Field)
		The original ending GPS Longitude (X) location coordinate point.

		Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	X_StartOrig	("Longitude - Start", dtDouble, Unit-Based Field) The original starting GPS Longitude (X) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Y_EndOrig	("Latitude - End", dtDouble, Unit-Based Field) The original ending GPS Latitude (Y) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Y_StartOrig	("Latitude - Start", dtDouble, Unit-Based Field) The original starting GPS Latitude (Y) location coordinate point. Note: The unit of measure for GPS coordinates is 'DD'.
decimal?	Z_EndOrig	("Elevation - End", dtDouble, Unit-Based Field) The original ending GPS Elevation (Z) location coordinate point; aka Height or Altitude.
decimal?	Z_StartOrig	("Elevation - End", dtDouble, Unit-Based Field) The original starting GPS Elevation (Z) location coordinate point; aka Height or Altitude.

Foreign References		
Table Name	**Call Value**	**Definition**
TrendTestPoint	lstTrendTestPoint	Master table for General Trending Test Points

TrendDates (lstTrendDates)

General Trending Test Dates: the test dates and associated information for trend points; represents the Test Dates grid in the General Trending detail.

<table>
<tr><th>Data Type</th><th>Field</th><th>Field Information (GUI Reference, GUI Data Type, Size, Purpose)</th></tr>
<tr><td>DateTime</td><td><b>TestDate</b></td><td>("Test Date", dtDate, Required Field)
Enter the test date for the trend point.
<b>Note</b>: This is a unique identifier for each test date within a test point.</td></tr>
<tr><td>string</td><td><b>TrendTestPoint</b></td><td>("Trend Test Point", dtString, Required Field, Maximum Length 15)
The trend point identifier for the equipment.</td></tr>
<tr><td>bool?</td><td>Baseline</td><td>("Baseline", dtInteger, Default Value 0)
Identify whether the reading test date is a baseline measurement; choices are: 0, No, N, 1, Yes, Y.</td></tr>
<tr><td>TG3?</td><td>OnOffFlag</td><td>("Stream", dtInteger, Default Value Internal)
Select the stream that identifies the process stream condition when the trend point was measured; choices are:
<table>
<tr><th>Stream</th><th>Definition</th></tr>
<tr><td>Internal</td><td>the test point is located and measured internally within its owning equipment</td></tr>
<tr><td>External (Off)</td><td>the test point is located externally and measured while its owning equipment is off stream</td></tr>
<tr><td>External (On)</td><td>the test point is located externally and measured while its owning equipment is on stream</td></tr>
</table></td></tr>
</table>

string	TestedBy	("Tested By", dtString, Maximum Length 35) Enter the name of the person or company who performed the reading survey for the trend point.

Foreign Reference		
Table Name	**Call Value**	**Definition**
TrendRead	lstTrendRead	Readings Detail - test dates, distances, and readings
TrendTestDate	rfTrendTestPoint	Master table for General Trending; Trend Point details

TrendRead (lstTrendRead)

General Trending Readings: the readings for a trend point; represents the Reading values in the Readings grid in the General Trending detail.

Data Type	Field	Field Information (GUI Reference, GUI Data Type, Size, Purpose)
DateTime	**TestDate**	("Test Date", dtDate, Required Field) Enter the test date for the test point. **Note**: This is a unique identifier for each test date within a test point.
string	**TrendTestPoint**	("Trend Test Point", dtString, Required Field, Maximum Length 15) The trend point identifier for the equipment.
string	**ReadType**	("Reading Type", dtString, Required Field, Maximum Length 15) From a lookup list, select the type of reading being measured. Example: Coupons could be weight or dimensions (length, width, depth).
decimal?	Reading	("Reading", dtDouble) Enter the measurement or reading value for each reading type defined for a test date. Note: Represented by the ReadingsCSV parameter.

Foreign Reference		
Table Name	**Call Value**	**Definition**
TrendDates	rfTrendDates	General Trending Test Dates

Call Examples

Similar to the Asset API, a debug flag may be included in a call to echo the data in the response message for troubleshooting.

QueryGTTP

Call without Debug Information

```
<!-- The QueryGTTP function returns data about a specified subset of General Trending Test
    Points in a tabular format. Any four fields can be specified to limit the query results
    - in this example, results are only returned associated with Test Points for Vessel
    equipment.  The ReadingsCSV field specifies all the types of readings populated in a
    given GTTP's latest readings in an X=Y;X2=Y2 format - for example, "Before
    Vent=9;Current=7;Gas Rate=8". Any given GTTP can have zero or more readings, one per
    Test Type. Only readings from the latest Test Date are included in this summary. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com"
   xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
 <soapenv:Header/>
 <soapenv:Body>
   <met:QueryGTTP>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipType>Vessel</met:EquipType>
   </met:QueryGTTP>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <QueryGTTPResponse xmlns="http://metegrity.com">
    <QueryGTTPResult>true</QueryGTTPResult>
    <OutList xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
       xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
       <a:ArrayOfstring>
        <a:string>Plant ID</a:string>
        <a:string>Equipment Type</a:string>
        <a:string>Equipment Number</a:string>
        <a:string>Trend Test Point</a:string>
        <a:string>Description</a:string>
        <a:string>Trend Type</a:string>
        <a:string>Last Test Date</a:string>
        <a:string>Last Update</a:string>
        <a:string>Maximum Reading</a:string>
        <a:string>Minimum Reading</a:string>
        <a:string>Next Test Date (Calculated)</a:string>
        <a:string>Next Test Due Date</a:string>
        <a:string>Reading Unit</a:string>
        <a:string>Interval</a:string>
        <a:string>Interval Unit</a:string>
        <a:string>Test Date</a:string>
        <a:string>Baseline</a:string>
        <a:string>Comments</a:string>
        <a:string>Functional Location</a:string>
        <a:string>Stream</a:string>
        <a:string>Tested By</a:string>
        <a:string>ReadingsCSV</a:string>
       </a:ArrayOfstring>
       <a:ArrayOfstring>
        <a:string>Plant 1</a:string>
        <a:string>Pipe</a:string>
        <a:string>P-1102A</a:string>
        <a:string>P1102A-WC</a:string>
        <a:string>Description of WC ratio trend point</a:string>
        <a:string>W/C ratio</a:string>
        <a:string>12/06/2015</a:string>
        <a:string>06/07/2016</a:string>
        <a:string>12</a:string>
        <a:string>6</a:string>
        <a:string i:nil="true"/>
        <a:string>09/07/2015</a:string>
        <a:string>Grams</a:string>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string>12/06/2015</a:string>
        <a:string>True</a:string>
        <a:string>comments text</a:string>
        <a:string i:nil="true"/>
        <a:string>ExternalOff</a:string>
        <a:string>JSmith</a:string>
        <a:string/>
       </a:ArrayOfstring>
       <a:ArrayOfstring>
        <a:string>Plant 1</a:string>
        <a:string>Pipe</a:string>
        <a:string>P-1102A</a:string>
        <a:string>P1102A-V1</a:string>
        <a:string i:nil="true"/>
        <a:string>Pressure</a:string>
        <a:string>23/06/2016</a:string>
        <a:string>22/06/2016</a:string>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string i:nil="true"/>
        <a:string>23/06/2016</a:string>
        <a:string>True</a:string>
        <a:string/>
        <a:string i:nil="true"/>
        <a:string>ExternalOn</a:string>
        <a:string i:nil="true"/>
        <a:string>Before Vent=150;After Vent=125</a:string>
       </a:ArrayOfstring>
    </OutList>
    <moreLeft>false</moreLeft>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </QueryGTTPResponse>
 </s:Body>
</s:Envelope>
```

QueryGTTrendTypes

Call without Debug Information

```
<!-- This simple function returns all of the GT Test Types currently defined in the Database
    Site the connectivity information indicates. Each type includes a type name and a
    simple description; the two string arrays parallel each other, so TypeDescriptions[2]
    describes the type named TrendTypes[2]. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:QueryGTTrendType>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
   </met:QueryGTTrendType>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <QueryGTTrendTypeResponse xmlns="http://metegrity.com">
    <QueryGTTrendTypeResult>true</QueryGTTrendTypeResult>
        <TrendTypes xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
          <a:string>Temperature</a:string>
          <a:string>Pressure</a:string>
          <a:string>W/C ratio</a:string>
          <a:string>Coupons</a:string>
          <a:string>Vent Monitoring</a:string>
          <a:string>% wt.Loss</a:string>
          <a:string>Length</a:string>
        </TrendTypes>
        <TypeDescriptions
          xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
          xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
          <a:string>Degrees F</a:string>
          <a:string i:nil="true"/>
          <a:string>W/C ratio</a:string>
          <a:string>Corrosion Coupon sizes</a:string>
          <a:string>Annulus pipe vent monitoring</a:string>
          <a:string>Weight loss in grams</a:string>
          <a:string i:nil="true"/>
       </TypeDescriptions>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </QueryGTTrendTypeResponse>
 </s:Body>
</s:Envelope>
```

CreateGTTrendType

Call without Debug Information

```
<!-- The CreateGTTrendType function adds a new universal Trending Type to the database Site
    indicated. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:CreateGTTrendType>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:TrendType>CP</met:TrendType>
    <a:string>Cathodic Protection Potentials</a:string>
   </met:CreateGTTrendType>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <CreateGTTrendTypeResponse xmlns="http://metegrity.com">
    <CreateGTTrendTypeResult>true</CreateGTTrendTypeResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </CreateGTTrendTypeResponse>
 </s:Body>
</s:Envelope>
```

UpdateGTTrendType

Call without Debug Information

```
<!-- The UpdateGTTrendType function is used to change the name or description for an
    existing database Trend Type. If the name of a Trending Type is changed, it will be
    changed automatically within all the GTTPs that use it. The old name should be specified
    in TrendType, and the new name in NewTrendType. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:UpdateGTTrendType>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:TrendType>W/C ratio</met:TrendType>
    <met:newTypeName>WC ratio</met:newTypeName>
    <met:TypeDescription>Water/Condensate Ratio</met:TypeDescription>
   </met:UpdateGTTrendType>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <UpdateGTTrendTypeResponse xmlns="http://metegrity.com">
    <UpdateGTTrendTypeResult>true</UpdateGTTrendTypeResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </UpdateGTTrendTypeResponse>
 </s:Body>
</s:Envelope>
```

DeleteGTTrendType

Call without Debug Information

```
<!-- The DeleteGTTrendType function deletes an existing Trending Type from the universal
    list in a database Site. A Trending Type can not be deleted if it is used by existing
    readings; delete the readings first. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:DeleteGTTrendType>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:TrendType>Length</met:TrendType>
   </met:DeleteGTTrendType>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <DeleteGTTrendTypeResponse xmlns="http://metegrity.com">
    <DeleteGTTrendTypeResult>true</DeleteGTTrendTypeResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </DeleteGTTrendTypeResponse>
 </s:Body>
</s:Envelope>
```

CreateGTTP

Call without Debug Information

```
<!-- CreateGTTP creates a new GTTP, specifying its minimum and maximum readings, unit of
    measure and test dates. As with the CP functions, Interval is measured in days, weeks,
    months or years; the unit can be specified in the same parameter as the interval or as
    a separate parameter as shown below. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:CreateGTTP>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-1102A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:GTTP>GTTP-22</met:GTTP>
    <met:TrendType>W/C ratio</met:TrendType>
    <met:Description>Desc Text</met:Description>
    <met:MinRead>4</met:MinRead>
    <met:MaxRead>9</met:MaxRead>
    <met:ReadingUnit>grams</met:ReadingUnit>
    <met:NextTestDateDue>07/09/2015</met:NextTestDateDue>
    <met:RInterval>7 days</met:RInterval>
    <!--<met:RIntervalUnit>?</met:RIntervalUnit>-->
    <met:dateFmt>MM/dd/yyyy</met:dateFmt>
   </met:CreateGTTP>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <CreateGTTPResponse xmlns="http://metegrity.com">
    <CreateGTTPResult>true</CreateGTTPResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </CreateGTTPResponse>
 </s:Body>
```

```
</s:Envelope>
```

CreateGTTP (Values XML)

Call without Debug Information

```
<!-- As with most API functions, you can also specify input to CreateGTTP using Values XML.
    This allows the creation of multiple Test Points at once, and also allows you to specify
    GeoLocation coordinates and other data for the created test points (as the example
    below shows. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:CreateGTTP>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:Values>
    [Records]
        [Values]
        [Value field="rfEquipment.EquipNum"]P-1102A[/Value]
        [Value field="rfEquipment.PlantNum"]Plant 1[/Value]
        [Value field="rfEquipment.EquipType"]Pipe[/Value]
        [Value field="GTTP"]GTTP-23[/Value]
        [Value field="TrendType"]W/C ratio[/Value]
        [Value field="Description"]Desc Text[/Value]
        [Value field="MinRead"]4[/Value]
        [Value field="MaxRead"]9[/Value]
        [Value field="ReadingUnit"]grams[/Value]
        [Value fmt="MM/dd/yyyy" field="NextDate"]07/09/2015[/Value]
        [Value field="RInterval"]7 days[/Value]
        [Value field="rfLocGUID.LongitudeStart"]67[/Value]
        [Value field="rfLocGUID.LatitudeStart"]71[/Value]
        [/Values]
    [/Records]
    </met:Values>
   </met:CreateGTTP>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <CreateGTTPResponse xmlns="http://metegrity.com">
    <CreateGTTPResult>true</CreateGTTPResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </CreateGTTPResponse>
 </s:Body>
</s:Envelope>
```

UpdateGTTP

Call without Debug Information

```
<!-- UpdateGTTP allows you to change certain properties of a Test
 Point after it has been created. However, the Trending Type
 for a test point that has readings cannot be altered. -->
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:UpdateGTTP>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-1102A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:GTTP>GTTP-1</met:GTTP>
    <met:TrendType>WC ratio</met:TrendType>
    <met:Description>Desc Text Updated</met:Description>
    <met:MinRead>6</met:MinRead>
    <met:MaxRead>12</met:MaxRead>
    <met:ReadingUnit>grams</met:ReadingUnit>
    <met:NextTestDateDue>07/09/2015</met:NextTestDateDue>
    <met:RInterval>7 days</met:RInterval>
    <met:dateFmt>MM/dd/yyyy</met:dateFmt>
   </met:UpdateGTTP>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <UpdateGTTPResponse xmlns="http://metegrity.com">
    <UpdateGTTPResult>true</UpdateGTTPResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </UpdateGTTPResponse>
 </s:Body>
</s:Envelope>
```

DeleteGTTP

Call without Debug Information

```
<!-- This API call deletes an entire GTTP from the database. As usual, multiple GTTPs can
    be specified using Values XML if desired. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:DeleteGTTP>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-1102A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:GTTP>GTTP-1</met:GTTP>
   </met:DeleteGTTP>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <DeleteGTTPResponse xmlns="http://metegrity.com">
    <DeleteGTTPResult>true</DeleteGTTPResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </DeleteGTTPResponse>
 </s:Body>
</s:Envelope>
```

AddGTTPReadings

Call without Debug Information

```
<!-- This API function allows readings to be added to a GTTP. The reading values are
    specified in ReadingsCSV, separated by semicolons; the order of this array should match
    the read types listed in ReadTypesCSV, which shows the variables that each reading
    corresponds to. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:AddGTTPReadings>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-1102A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:GTTP>GTTP-1</met:GTTP>
    <met:TestDate>04/09/2014</met:TestDate>
    <met:Baseline>T</met:Baseline>
    <met:Stream>external (off)</met:Stream>
    <met:TestedBy>julian</met:TestedBy>
    <met:Comments>comments text</met:Comments>
    <met:ReadingsCSV>4.4;5.5;6.6;7.7;8.8</met:ReadingsCSV>
    <met:ReadTypesCSV>Cond Rate;Gas Rate;Gas:H2S;Current;Length</met:ReadTypesCSV>
    <met:dateFmt>MM/dd/yyyy</met:dateFmt>
   </met:AddGTTPReadings>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <AddGTTPReadingsResponse xmlns="http://metegrity.com">
    <AddGTTPReadingsResult>true</AddGTTPReadingsResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </AddGTTPReadingsResponse>
 </s:Body>
</s:Envelope>
```

AddGTTPReadings (Values XML)

Call without Debug Information

```
<!-- GTTP readings can also be specified using Values XML, which allows more than one Test
    Date to be added to a single Test Point with one call, or more than one Test Point to
    have readings added. The XML syntax is shown below. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:AddGTTPReadings>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:Values>
    [Records]
        [Values]
        [Value field="rfEquipment.EquipNum"]V-1102A[/Value]
        [Value field="rfEquipment.PlantNum"]Plant 1[/Value]
        [Value field="rfEquipment.kvEquipType"]Vessel[/Value]
        [Value field="GTTP"]GTTP-1[/Value]
        [Value field="lstTrendDates[1].TestDate"]06/03/2015[/Value]
        [Value field="lstTrendDates[1].Baseline"]T[/Value]
        [Value field="lstTrendDates[1].Stream"]External (Off)[/Value]
        [Value field="lstTrendDates[1].TestedBy"]Julian[/Value]
        [Value field="lstTrendDates[1].Comments"]comments text[/Value]
        [Value field="lstTrendDates[1].ReadingsCSV"]4.4;5.5;6.6;7.7;8.8[/Value]
        [Value field="lstTrendDates[1].ReadTypesCSV"]Cond Rate;Gas
       Rate;Gas:H2S;Current;Length[/Value]
        [/Values]
    [/Records]
    </met:Values>
    <met:dateFmt>MM/dd/yyyy</met:dateFmt>
   </met:AddGTTPReadings>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <AddGTTPReadingsResponse xmlns="http://metegrity.com">
    <AddGTTPReadingsResult>true</AddGTTPReadingsResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </AddGTTPReadingsResponse>
 </s:Body>
</s:Envelope>
```

EditGTTPReadings

Call without Debug Information

```
<!-- EditGTTPReadings alters existing readings for a given test point and test date. Other
    than affecting existing readings, it works identically to the AddGTTPReadings function.
    -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
```

```
 <soapenv:Header/>
 <soapenv:Body>
   <met:EditGTTPReadings>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-1102A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:GTTP>GTTP-1</met:GTTP>
    <met:TestDate>04/07/2014</met:TestDate>
    <met:Baseline>T</met:Baseline>
    <met:Stream>external (off)</met:Stream>
    <met:TestedBy>julian</met:TestedBy>
    <met:Comments>comments text edited</met:Comments>
    <met:ReadingsCSV>;50.5;;70.7;</met:ReadingsCSV>
    <met:ReadTypesCSV>Cond Rate;Gas Rate;Gas:H2S;Current;Length</met:ReadTypesCSV>
    <met:dateFmt>MM/dd/yyyy</met:dateFmt>
   </met:EditGTTPReadings>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <EditGTTPReadingsResponse xmlns="http://metegrity.com">
    <EditGTTPReadingsResult>true</EditGTTPReadingsResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </EditGTTPReadingsResponse>
 </s:Body>
</s:Envelope>
```

DeleteGTTPReadings

Call without Debug Information

```
<!-- This API call deletes a specific Test Date worth of readings from a specified GTTP.
    Using Values XML, more than one test date can be deleted, or readings can be deleted
    from more than one Test Point. -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:met="http://metegrity.com">
 <soapenv:Header/>
 <soapenv:Body>
   <met:DeleteGTTPReadings>
    <met:SiteName>[Visions SiteName]</met:SiteName>
    <met:UserName>[Visions UserName]</met:UserName>
    <met:Password>[Visions Password]</met:Password>
    <met:EquipNum>V-1102A</met:EquipNum>
    <met:PlantNum>Plant 1</met:PlantNum>
    <met:EquipType>Vessel</met:EquipType>
    <met:GTTP>GTTP-1</met:GTTP>
    <met:TestDate>04/08/2014</met:TestDate>
    <met:dateFmt>MM/dd/yyyy</met:dateFmt>
   </met:DeleteGTTPReadings>
 </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
   <DeleteGTTPReadingsResponse xmlns="http://metegrity.com">
    <DeleteGTTPReadingsResult>true</DeleteGTTPReadingsResult>
    <ErrorMsg>Operation successful.</ErrorMsg>
   </DeleteGTTPReadingsResponse>
 </s:Body>
</s:Envelope>
```

Additional Information

Visions Enterprise Help File

Lookup Data > F3 Lookup Lists

Trending > General Trending