



# P2000

Security Management System

---

## XML-RPC

Application Programming Interface

---

PRELIMINARY



# P2000

## Security Management System

# XML-RPC

## Application Programming Interface

Version 3.11 and higher, October, 2011

24-10618-171 Revision –



ELIMINARY

Copyright 2011  
Johnson Controls, Inc.  
All Rights Reserved

No part of this document may be reproduced without the prior permission of Johnson Controls, Inc.

### **Acknowledgment**

Cardkey P2000, BadgeMaster, and Metasys are trademarks of Johnson Controls, Inc.

All other company and product names are trademarks or registered trademarks of their respective owners.

If this document is translated from the original English version by Johnson Controls, Inc., all reasonable endeavors will be used to ensure the accuracy of translation. Johnson Controls, Inc. shall not be liable for any translation errors contained herein or for incidental or consequential damages in connection with the furnishing or use of this translated material.

Due to continuous development of our products, the information in this document is subject to change without notice. Johnson Controls, Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with furnishing or use of this material. Contents of this publication may be preliminary and/or may be changed at any time without any obligation to notify anyone of such revision or change, and shall not be regarded as a warranty.

PRELIMINARY



# TABLE OF CONTENTS

<b>Overview</b>	1
<b>Details</b>	2
XML_RPC Calling Convention	2
Authentication	2
Security	3
Scheduled Actions	3
Troubleshooting	3
<b>API Reference</b>	4
Return Values	4
Method List	7
Access Group Methods	7
Alarm Methods	10
Badge Methods	13
Cardholder Methods	18
Company Methods	24
Counter Methods	25
Department Methods	27
Door Methods	28
Event Methods	41
Heart Beat Methods	43
Input Methods	44
Output Methods	47
P2000 Methods	53
Panel Methods	53
Partition Methods	57
Terminal Methods	58
Timezone Methods	59
UDF Methods	63
XAction Methods	65

PRELIMINARY



# P2000 XML-RPC INTERFACE

This document describes the Application Programming Interface (API) for the P2000 XML-RPC Interface. This API details the data and interface requirements for both P2000 applications and third-party applications to call methods in the P2000 XML-RPC Interface. This document only covers the P2000 V3.x series.

## NOTE

*This document is intended to be used by programmers or other qualified professionals who possess a reasonable level of experience with application program writing.*

## OVERVIEW

The P2000 XML-RPC is a P2000 interface that allows applications to execute methods in the P2000 system to lock/unlock doors, set/reset output points, acknowledge alarms, trigger P2000 host events and more. XML-RPC is a method of remote procedure calling using HTTP as the transport and XML as the encoding.

XML-RPC is an open standard designed to be as simple as possible, while allowing complex data structures to be transmitted, processed, and returned. It can be used with Perl, Java, Python, C, C++, .Net, and many other programming languages. Many different implementations are freely available for UNIX, Windows, and Macintosh systems. For C++ applications, Johnson Controls recommends the XmlRpc++ implementation by Chris Morley. The full XML-RPC spec can be viewed at [www.xmlrpc.com](http://www.xmlrpc.com).

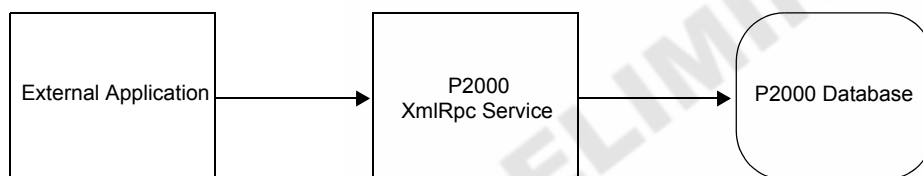


Figure 1: XML-RPC Interface Operation

## DETAILS

### XML\_RPC Calling Convention

An XML-RPC message is an HTTP-POST request. The body of the request is in XML. A procedure executes on the server and the value it returns is also formatted in XML. An example XML-RPC message follows:

```
POST /RPC2 HTTP/1.1
User-Agent: XMLRPC++ 0.7
Host: p2000:41023
Content-Type: text/xml
Content-length: 220

<?xml version="1.0"?>
<methodCall><methodName>SetOutput</methodName>
  <params>
    <param>
      <value>Cardkey</value>
    </param>
    <param>
      <value>bWFzdGVy</value>
    </param>
    <param>
      <value>{E9F52C29-193F-4F0D-885D-87A6DC82992E}</value>
    </param>
    <param>
      <value><i4>0</i4></value>
    </param>
  </params>
</methodCall>
```

### Authentication

The first two parameters to all methods of the P2000 XML-RPC interface are the username and password. The username must be either a valid P2000 operator username or a valid P2000 cardholder name. A P2000 cardholder name must be in the same form as required by P2000 Web Access, where the username is a combined value of the first name and the last name separated by a period. For example, the username for “John Smith” would be “john.smith.”

The password parameter has several options related to how it is encrypted. This setting is the same for all applications using the P2000 XML-RPC Interface on a P2000 system. The password mode is set on the *XmlRpc* tab of the *Edit Site Parameters* application of the P2000 software. If the password mode is set to “Ignore,” the password parameter to all XML-RPC methods is ignored.

## Security

The P2000 XML-RPC Service will only accept messages from a list of valid IP addresses and valid computers. The list of valid IP addresses always includes the IP address of the P2000 server. The list of valid IP addresses and computer names is configured in the *External IP* section of the P2000 *System Configuration* application. These computer names are converted to valid IP addresses by calling the “gethostbyname” function of Windows. Note that this conversion from computer name to IP address is only performed when the XML-RPC service starts.

## Scheduled Actions

When commands are scheduled (such as DoorUnlockScheduled), the P2000 uses the same mechanism for executing these commands as it does for P2000 Host Events with delayed actions. These actions are performed by the P2000 RTL Route Service and should typically be executed within 10 seconds of their scheduled time. Since the P2000 software was not designed for large numbers of scheduled actions, the use of scheduled commands should be kept to a minimum.

## Troubleshooting

The P2000 XML-RPC service can be ran in “debug” mode to see additional information about commands that are being processed. First stop the XML-RPC service using Windows or P2000 Service Control. Then open a command prompt in the P2000 “bin” directory and start the XML-RPC service in debug mode with the following command:

```
P2KXmlRpc -d
```

The XML-RPC service will now operate as a console application and will output information to its console window. The XML-RPC service can be stopped when running in debug mode by entering a <Ctrl> C in its console window.

## API REFERENCE

The P2000 XML-RPC methods documented below are described using C/C++ syntax. The arguments listed are actually sent as a list of param values as show in “XML\_RPC Calling Convention” on page 2.

### Return Values

Most P2000 XML-RPC methods return their execution results in the same fashion. Methods that return a BOOL value will return the following XML data when successful:

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><i4>1</i4></value>
    </param>
  </params>
</methodResponse>
```

When methods fail, they will return a “fault” structure that contains an integer “faultCode” and a “faultString.” A typical fault will return XML data similar to the following:

```
<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><i4>-1</i4></value>
        </member>
        <member>
          <name>faultString</name>
          <value>type error</value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

The following is the list of possible fault codes:

- 1 Unknown error
- 2 Error opening database
- 3 Error reading from or writing to database
- 4 Invalid username or password
- 5 Unexpected error during processing
- 6 Invalid name
- 7 Duplicate name
- 8 Partition not found in database
- 9 Access Group not found in database

- 10 Terminal not found in database
- 11 Terminal Group not found in database
- 12 Timezone not found in database
- 13 Timezone is in use
- 14 Alarm not found in database
- 15 Alarm is in incorrect state
- 16 Event not found in database
- 17 Scheduled action not found in database
- 18 Badge not found in database
- 19 Cardholder not found in database
- 20 Invalid UDF type
- 21 Counter not found in database
- 22 Error downloading data to controller
- 23 Terminal is not a reader
- 24 Terminal is disabled
- 25 Input not found in database
- 26 Output not found in database
- 27 Panel not found in database
- 28 Output Group not found in database
- 29 Invalid security level
- 30 Alarm already exists
- 31 Invalid UDF selection choice
- 32 UDF not found in database
- 33 Operation not supported

Methods that return data, such as DoorGetScheduled, return an array of structures where each structure contains a scheduled command. For example:

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value>
        <array>
          <data>
            <value>
              <struct>
                <member>
                  <name>ScheduledAction</name>
                  <value>DoorTimedUnlock</value>
                </member>
                <member>
                  <name>ItemName</name>
                  <value>Front Lobby Door</value>
                </member>
                <member>
                  <name>ItemGuid</name>
                  <value>{8AFBC4D6-9185-43BD-9434-3C655E0A8D17}</value>
                </member>
              </struct>
            </value>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodResponse>
```

```

        <member>
            <name>ScheduledID</name>
            <value>915</value>
        </member>
        <member>
            <name>ScheduledTime</name>
            <value><dateTime.iso8601>01050911T09:43:00</dateTime.iso8601></value>
        </member>
        <member>
            <name>ScheduledParam</name>
            <value><i4>60</i4></value>
        </member>
    </struct>
</value>
<value>
    <struct>
        <member>
            <name>ScheduledAction</name>
            <value>DoorTimedUnlock</value>
        </member>
        <member>
            <name>ItemName</name>
            <value>Engineering Lab Door</value>
        </member>
        <member>
            <name>ItemGuid</name>
            <value>{ABE254EB-38CA-4C06-8E2A-F6C6A21E64B3}</value>
        </member>
        <member>
            <name>ScheduledID</name>
            <value>951</value>
        </member>
        <member>
            <name>ScheduledTime</name>
            <value><dateTime.iso8601>01050911T10:13:00</dateTime.iso8601></value>
        </member>
        <member>
            <name>ScheduledParam</name>
            <value><i4>35</i4></value>
        </member>
    </struct>
</value>
</data>
</array>
</value>
</param>
</params>
</methodResponse>

```

## Method List

### Access Group Methods

**AccessGroupAdd(string sUserName, string sPassword, string sAccessGroupName, string sAccessGroupGuid, string sPartitionGuid, int iPublic, string array sTerminalGuids, string array sTerminalGroupGuids)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sAccessGroupName	name for new Access Group
sAccessGroupGuid	GUID for new Access Group (autoassigned if "NULL")
sPartitionGuid	GUID for the partition which owns this access group (ignored on non-partitioned systems)
iPublic	0 = non-public, 1 = public (ignored on non-partitioned systems)
sTerminalGuids	array of Terminal GUIDs
sTerminalGroupGuids	array of Terminal Group GUIDs

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

**AccessGroupAddByName(string sUserName, string sPassword, string sAccessGroupName, string sAccessGroupGuid, string sPartitionName, int iPublic, string array sTerminals, string array sTerminalGroups)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sAccessGroupName	name for new Access Group
sAccessGroupGuid	GUID for new Access Group (autoassigned if "NULL")
sPartitionName	name of the partition which owns this access group (ignored on non-partitioned systems)
iPublic	0 = non-public, 1 = public (ignored on non-partitioned systems)
sTerminals	array of Terminal names
sTerminalGroups	array of Terminal Group names

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

**AccessGroupGetList(string sUserName, string sPassword, string sPartitionGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of Partition to get AccessGroup list for (ignored on non-partitioned systems)

Returns an array of structure with the following members:

Member	Description
ItemName	AccessGroup name
ItemGuid	AccessGroup Global Unique Identifier
PartitionName	name of partition which owns this AccessGroup
PartitionGuid	Guid of partition which owns this AccessGroup
Public	0 = non-public, 1 = public

Available in P2000 V3.6 SP1 and later.

**AccessGroupGetListByName(string sUserName, string sPassword, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	name of Partition to get AccessGroup list for (ignored on non-partitioned systems)

Returns an array of structure with the following members:

Member	Description
ItemName	AccessGroup name
ItemGuid	AccessGroup Global Unique Identifier
PartitionName	name of partition which owns this AccessGroup
PartitionGuid	Guid of partition which owns this AccessGroup
Public	0 = non-public, 1 = public

Available in P2000 V3.6 SP1 and later.

**AccessGroupModify(string sUserName, string sPassword, string sAccessGroupGuid, string array sTerminalGuids, string array sTerminalGroupGuids)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sAccessGroupGuid	GUID of Access Group
sTerminalGuids	array of Terminal GUIDs
sTerminalGroupGuids	array of Terminal Group GUIDs

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.



**AccessGroupModifyByName(string sUserName, string sPassword, string sAccessGroupName, string sPartitionName, string array sTerminalNames, string array sTerminalGroupNames)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sAccessGroupName	name of Access Group
sPartitionName	name of partition (ignored on non-partitioned systems)
sTerminalNames	array of Terminal names
sTerminalGroupNames	array of Terminal Group names

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**AccessGroupDelete(string sUserName, string sPassword, string sAccessGroupGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sAccessGroupGuid	GUID of Access Group

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.5 and later.

**Note:** The access group will be deleted from all badges that are assigned this access group.

**AccessGroupDeleteByName(string sUserName, string sPassword, string sAccessGroupName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sAccessGroupName	name of Access Group
sPartitionName	name of partition (ignored on non-partitioned systems)

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**Note:** The access group will be deleted from all badges that are assigned this access group.

## Alarm Methods

**AlarmCreate(string sUserName, string sPassword, string sAlarmTypeGuid, string sPartitionGuid, int iPublic, string sItemName, string sDescription, string sInstructions, int iPriority, string sCategory, string sQueryString, int iAckRequired, int iResponseRequired, int iPopup)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sAlarmTypeGuid	Unique type GUID of alarm to be created (alarms with the same type GUID are considered to be alarms for the same item)
sPartitionGuid	Guid of partition that alarm belongs to (ignored on non-partitioned systems)
iPublic	Public flag of alarm (ignored on non-partitioned systems)
sItemName	Name of item generating alarm
sDescription	Alarm description
sInstructions	Alarm Instructions (may be left blank)
iPriority	Alarm priority (0 to 255)
sCategory	Alarm category (default category if blank)
sQueryString	Alarm query string (may be left blank)
iAckRequired	Ack required flag
iResponseRequired	Response required flag
iPopup	Alarm Monitor popup flag

Returns the Guid of the alarm created if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**AlarmCreateDetailed(string sUserName, string sPassword, string sAlarmTypeGuid, string sPartitionGuid, int iPublic, string sItemName, string sDescription, string sInstructions, int iAlarmType, int iPriority, string sCategory, string sQueryString, int iAckRequired, int iResponseRequired, int iPopup, datetime dtAlarmTime, int iItemState, int iCompletionState)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sAlarmTypeGuid	Unique type GUID of alarm to be created (alarms with the same type GUID are considered to be alarms for the same item)
sPartitionGuid	Guid of partition that alarm belongs to (ignored on non-partitioned systems)
iPublic	Public flag of alarm (ignored on non-partitioned systems)
sItemName	Name of item generating alarm
sDescription	Alarm description
sInstructions	Alarm Instructions (may be left blank)
iAlarmType	P2000 Alarm Type (Generic Alarm = 1)
iPriority	Alarm priority (0 to 255)
sCategory	Alarm category (default category if blank)
sQueryString	Alarm query string (may be left blank)
iAckRequired	Ack required flag
iResponseRequired	Response required flag
iPopup	Alarm Monitor popup flag
dtAlarmTime	Time of the alarm in UTC
iItemState	Current item state (-1 = ignore, 0 = secure, 1 = alarm)
iCompletionState	Item state required for alarm completion (-1 = ignore, 0 = secure, 1 = alarm)

Returns the Guid of the alarm created if successful, Fault Structure if failure.  
Available in P2000 V3.8 SP2 and later.

**AlarmGetList(string sUserName, string sPassword, string sPartitionGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of partition to search (ignored on non-partitioned systems)

Returns a structure with the following members for each alarm:

Member	Description
ItemName	Description of alarm
ItemGuid	Guid of alarm
PartitionGuid	Guid of partition which owns this alarm
PartitionName	name of partition which owns this alarm
Public	0 = non-public, 1 = public
AlarmInstruction	Instruction text for alarm
AlarmType	Type of alarm (see below)
AlarmTypeID	Type ID of alarm
AlarmTypeGuid	Type Guid of alarm
AlarmCategory	Category of alarm
AlarmPriority	Priority of alarm
AlarmEscalationLevel	Escalation Level of alarm
AlarmState	1 = completed, 2 = responding, 3 = acknowledged, 4 = pending
AlarmStateTime	Time that alarm went to current state
AlarmItemState	State of item generating alarm
AlarmItemStateTime	Time that item went to current state
AlarmCompletionItemState	State of item required to complete alarm
AlarmAckRequired	1 = alarm requires acknowledge before completion
AlarmResponseRequired	1 = alarm requires response before completion

Alarm Types:

GENERIC	1
PANEL_INPUT_POINT	2
AREA	3
GUARDTOUR	4
MUSTER_RUNNING	5
MUSTER_ZONE_STATUS	6
MUSTER_WHEN_DISABLED	7
MUSTER_ABORTED	8
LOOP_TAMPER	9
EVENT	10
G3_EVENT	11
AV_MOTION	12
AV_BEHAVIOR	13
AV_VIDEOLOSS	14
AV_DRYCONTACT	15
AV_SYSTEM	16
INTRUSION_ZONE	17

Available in P2000 V3.8 and later.

**AlarmGetListByName(string sUserName, string sPassword, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	Name of partition to search (ignored on non-partitioned systems)

Returns a structure with the following members for each alarm:

Member	Description
ItemName	Description of alarm
ItemGuid	Guid of alarm
PartitionGuid	Guid of partition which owns this alarm
PartitionName	name of partition which owns this alarm
Public	0 = non-public, 1 = public
AlarmInstruction	Instruction text for alarm
AlarmType	Type of alarm (see below)
AlarmTypeID	Type ID of alarm
AlarmTypeGuid	Type Guid of alarm
AlarmCategory	Category of alarm
AlarmPriority	Priority of alarm
AlarmEscalationLevel	Escalation Level of alarm
AlarmState	1 = completed, 2 = responding, 3 = acknowledged, 4 = pending
AlarmStateTime	Time that alarm went to current state
AlarmItemState	State of item generating alarm
AlarmItemStateTime	Time that item went to current state
AlarmCompletionItemState	State of item required to complete alarm
AlarmAckRequired	1 = alarm requires acknowledge before completion
AlarmResponseRequired	1 = alarm requires response before completion

Alarm Types:

GENERIC	1
PANEL_INPUT_POINT	2
AREA	3
GUARDTOUR	4
MUSTER_RUNNING	5
MUSTER_ZONE_STATUS	6
MUSTER_WHEN_DISABLED	7
MUSTER_ABORTED	8
LOOP_TAMPER	9
EVENT	10
G3_EVENT	11
AV_MOTION	12
AV_BEHAVIOR	13
AV_VIDEOLOSS	14
AV_DRYCONTACT	15
AV_SYSTEM	16
INTRUSION_ZONE	17

Available in P2000 V3.8 and later.

**AlarmGetState(string sUserName, string sPassword, string sAlarmGuid))**

Argument	Description
UserName	valid P2000 username
sPassword	valid P2000 password
sAlarmGuid	database GUID of alarm to be updated

Returns a structure with the following members for the specified alarm State  
 0 = not found, 1 = completed, 2 = responding, 3 = acknowledged, 4 = pending

Available in P2000 V3.5 and later.

**AlarmUpdate(string sUserName, string sPassword, string sAlarmGuid, int iAlarmState, string sResponse)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sAlarmGuid	P2000 V3.x: database GUID of alarm to be updated
iAlarmState	1 = alarm completed, 2 = alarm responding, 3 = alarm acknowledged
sResponse	response to be recorded with alarm (only valid for iAlarmState = 2)

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

**AlarmUpdateItemState(string sUserName, string sPassword, string sAlarmGuid, int itemState)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sAlarmGuid	P2000 V3.x: database GUID of alarm to be updated
itemState	Current item state (-1 = ignore, 0 = secure, 1 = alarm)

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.8 SP4 and later.

**Badge Methods****BadgeAddAccessGroup(string sUserName, string sPassword, string sBadgeNumber, string sAccessGroupGuid, string sTimezoneGuid, [datetime dtStartDate, datetime dtVoidDate, string sSiteName])**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sBadgeNumber	Badge number
sAccessGroupGuid	GUID of Access Group
sTimezoneGuid	GUID of Timezone

Optional arguments in P2000 V3.9 and later:

Argument	Description
dtStartDate	date Access Group starts (NULL if 0:0:00 AM Jan 1, 1900)
dtVoidDate	date Access Group ends (NULL if 0:0:00 AM Jan 1, 1900)
sSiteName	name of Enterprise site to modify (ignored for non-Enterprise systems)

Returns TRUE (1) if successful, FALSE (0) if badge already contains Access Group and Timezone pair, Fault Structure if failure.

Access Group and Timezone are added to the first unused Access Group and Timezone pair of the badge.

Available in P2000 V3.6 SP1 and later.

**BadgeAddAccessGroupByName(string sUserName, string sPassword, string sBadgeNumber, string sPartitionName, string sAccessGroupName, string sTimezoneName[datetime dtStartDate, datetime dtVoidDate, string sSiteName])**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sBadgeNumber	Badge number
sPartitionName	name of partition (ignored on non-partitioned systems)
sAccessGroupName	name of Access Group
sTimezoneName	name of Timezone

Optional arguments in P2000 V3.9 and later:

Argument	Description
dtStartDate	date Access Group starts (NULL if 0:0:00 AM Jan 1, 1900)
dtVoidDate	date Access Group ends (NULL if 0:0:00 AM Jan 1, 1900)
sSiteName	name of Enterprise site to modify (ignored for non-Enterprise systems)

Returns TRUE (1) if successful, FALSE (0) if badge already contains Access Group and Timezone pair, Fault Structure if failure.

Access Group and Timezone are added to the first unused Access Group and Timezone pair of the badge.

Available in P2000 V3.8 SP2 and later.

**BadgeGetInfo(string sUserName, string sPassword, string sBadgeNumber, string sSiteName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sBadgeNumber	Badge number

Optional arguments in P2000 V3.9 and later:

Argument	Description
sSiteName	name of Enterprise site to get data for (ignored for non-Enterprise systems)

Returns a structure with the following members for the specified badge:

Member	Description
PartitionGuid	Guid of partition which owns this badge
PartitionName	name of partition which owns this badge
Public	0 = non-public, 1 = public
CardholderDatabaseID	Cardholder database ID
StartTimestamp	Start date/time
ExpireTimestamp	Expire date/time
BadgeNumber	(P2000 V3.9 and later only)
FacilityCode	(P2000 V3.9 and later only)
IssueLevel	Level of issue
Purpose	configured badge purpose
Reason	configured badge reason
SecurityLevel	configured security level of badge
Disabled	Disable flag
Trace	Trace flag
EventPrivilege	Event privilege level
ExecutivePriv	(P2000 V3.9 and later only)
Override	Override flag

Available in P2000 V3.6 and later.

**BadgeGetInfoDetailed(string sUserName, string sPassword, string sBadgeNumber, string sSiteName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sBadgeNumber	Badge number

Optional arguments in P2000 V3.9 and later:

Argument	Description
sSiteName	name of Enterprise site to get data for (ignored for non-Enterprise systems)

Returns a structure with the following members for the specified badge:

Member	Description
PartitionGuid	Guid of partition which owns this badge
PartitionName	name of partition which owns this badge
Public	0 = non-public, 1 = public
CardholderFirstName	Cardholder first name
CardholderMiddleName	Cardholder middle name
CardholderLastName	Cardholder last name
CardholderGuid	Cardholder Global Unique Identifier
CardholderDatabaseID	Cardholder database ID
StartTimestamp	Start date/time
ExpireTimestamp	Expire date/time
BadgeNumber	(P2000 V3.9 and later only)
FacilityCode	(P2000 V3.9 and later only)
IssueLevel	Level of issue

Member	Description
PINCode	(P2000 V3.9 and later only)
Description	(P2000 V3.9 and later only)
Purpose	configured badge purpose
Reason	configured badge reason
UseEnterpriseRights	(P2000 V3.9 and later only)
SecurityLevel	configured security level of badge
GuardTourPriority	(P2000 V3.9 and later only)
Disabled	Disable flag
Trace	Trace flag
EventPrivilege	Event privilege level
ExecutivePriv	(P2000 V3.9 and later only)
Override	Override flag
FlagA	(P2000 V3.9 and later only)
FlagB	(P2000 V3.9 and later only)
FlagC	(P2000 V3.9 and later only)
AccessGroups	an array of AccessGroup structures

An AccessGroup structure contains the following members:

Member
AccessGroupName
AccessGroupGuid
TimezoneGuid
TimezoneName
StartDate (P2000 V3.9 and later only)
VoidDate (P2000 V3.9 and later only)

Available in P2000 V3.6 and later.

#### **BadgeGetLastBadgeInfo(string sUserName, string sPassword, string sBadgeNumber)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sBadgeNumber	Badge number

Returns a structure with the following members for the specified badge:

Member	Description
LastValidTerminal	name of Terminal where badge was last granted access
LastValidTerminalGuid	Guid of Terminal where badge was last granted access
LastValidTime	date/time when badge was last granted access
LastInvalidTerminal	name of Terminal where badge was last denied access
LastInvalidTerminalGuid	Guid of Terminal where badge was last denied access
LastInvalidTime	date/time when badge was last denied access
EntryExitStatus	current status of badge (In, Out, or Undefined)

Available in P2000 V3.7 and later.



**BadgeGetPurpose(string sUserName, string sPassword, string sBadgeNumber)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sBadgeNumber	Badge number

Returns a structure with the following member for the specified badge:

Member	Description
BadgePurpose	configured badge purpose

Available in P2000 V3.6 and later.

**BadgeRemoveAccessGroup(string sUserName, string sPassword, string sBadgeNumber, string sAccessGroupGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sBadgeNumber	Badge number
sAccessGroupGuid	GUID of Access Group

Optional arguments in P2000 V3.9 and later:

Argument	Description
sSiteName	name of Enterprise site to get data for (ignored for non-Enterprise systems)

Returns TRUE (1) if successful, FALSE (0) if badge did not contain Access Group and Timezone pair, Fault Structure if failure.

Available in P2000 V3.6 SP1 and later.

**BadgeRemoveAccessGroupByName(string sUserName, string sPassword, string sBadgeNumber, string sPartitionName, string sAccessGroupName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sBadgeNumber	Badge number
sPartitionName	name of partition (ignored on non-partitioned systems)
sAccessGroupName	name of Access Group

Optional arguments in P2000 V3.9 and later:

Argument	Description
sSiteName	name of Enterprise site to get data for (ignored for non-Enterprise systems)

Returns TRUE (1) if successful, FALSE (0) if badge did not contain Access Group and Timezone pair, Fault Structure if failure.

Available in P2000 V3.6 SP1 and later.

**BadgeSetEntryExitStatus(string sUserName, string sPassword, string sBadgeNumber, int iStatus)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sBadgeNumber	Badge number
iStatus	0 = undefined, 1 = in, 2 = out

Returns TRUE (1) if successful, FALSE (0) if badge does not exist, Fault Structure if failure.  
Available in P2000 V3.8 SP2 and later.

**Cardholder Methods****CardholderGetAllUdf(string sUserName, string sPassword, string sCardholderGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sCardholderGuid	Guid of cardholder

Returns a structure with the following members for the specified cardholder:

Member	Description
ItemName	Field name
ItemValue	Value of a field
ItemGuid	Field Global Unique Identifier

Available in P2000 V3.6 and later.

**CardholderGetBadgeList(string sUserName, string sPassword, string sCardholderGuid, string sPartitionGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sCardholderGuid	Guid of cardholder
sPartitionGuid	Guid of partition to search (ignored on non-partitioned systems)

Returns an array of structures with the following member for each badge owned by this cardholder:

Member	Description
BadgeNumber	Badge number

Available in P2000 V3.6 SP1 and later.

**CardholderGetImage(string sUserName, string sPassword, string sCardholderGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sCardholderGuid	Guid of cardholder

Returns a structure with the following member for the specified cardholder:

Member	Description
CardholderImage	Cardholder image (Base64 encoded)

Available in P2000 V3.8 and later.

**CardholderGetImageByID(string sUserName, string sPassword, string sEmployeeID)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sEmployeeID	employee ID of cardholder

Returns a structure with the following member for the specified cardholder:

Member	Description
CardholderImage	Cardholder image (Base64 encoded)

Available in P2000 V3.8 and later.

**CardholderGetInfo(string sUserName, string sPassword, string sCardholderGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sCardholderGuid	Guid of cardholder

Returns a structure with the following members for the specified cardholder:

Member	Description
PartitionGuid	Guid of partition which owns this cardholder
PartitionName	name of partition which owns this cardholder
Public	0 = non-public, 1 = public
CardholderFirstName	Cardholder first name
CardholderMiddleName	Cardholder middle name
CardholderLastName	Cardholder last name
CardholderGuid	Cardholder Global Unique Identifier
CardholderDatabaseID	Cardholder database ID
CardholderAddress	Cardholder address
CardholderPhone	Cardholder phone
CardType	Card type
DeptName	Department
CompanyName	Company
EmployeeID	Employee ID
Email	Cardholder's email
StartTimestamp	Start date/time of a cardholder
ExpireTimestamp	Expire date/time of a cardholder
WebAccessGroup	WebAccess Menu Permission Group

Available in P2000 V3.6 and later.

**CardholderGetInfoByID(string sUserName, string sPassword, string sEmployeeID)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sEmployeeID	employee ID of cardholder

Returns a structure with the following members for the specified cardholder:

Member	Description
PartitionGuid	Guid of partition which owns this cardholder
PartitionName	name of partition which owns this cardholder
Public	0 = non-public, 1 = public
CardholderFirstName	Cardholder first name
CardholderMiddleName	Cardholder middle name
CardholderLastName	Cardholder last name
CardholderGuid	Cardholder Global Unique Identifier
CardholderDatabaseID	Cardholder database ID
CardholderAddress	Cardholder address
CardholderPhone	Cardholder phone
CardType	Card type
DeptName	Department
CompanyName	Company
EmployeeID	Employee ID
Email	Cardholder's email
StartTimestamp	Start date/time of a cardholder
ExpireTimestamp	Expire date/time of a cardholder
WebAccessGroup	WebAccess Menu Permission Group

Available in P2000 V3.6 SP1 and later.

**CardholderGetLastBadgeInfo(string sUserName, string sPassword, string sEmployeeID)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sEmployeeID	employee ID of cardholder

Returns a structure with the following members for the specified badge:

Member	Description
LastValidBadge	badge number of Cardholder that was last granted access
LastValidTerminal	name of Terminal where badge was last granted access
LastValidTerminalGuid	Guid of Terminal where badge was last granted access
LastValidTime	date/time when badge was last granted access
LastInvalidBadge	badge number of Cardholder that was last denied access
LastInvalidTerminal	name of Terminal where badge was last denied access
LastInvalidTerminalGuid	Guid of Terminal where badge was last denied access
LastInvalidTime	date/time when badge was last denied access
EntryExitStatus	current status of badge (In, Out, or Undefined)

Available in P2000 V3.7 and later.

**Note:** When possible, use BadgeGetLastBadgeInfo since it is significantly more efficient.

**CardholderGetLastBadgeInfoByID(string sUserName, string sPassword, string sCardholderGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sCardholderGuid	Guid of cardholder

Returns a structure with the following members for the specified badge:

Member	Description
LastValidBadge	badge number of Cardholder that was last granted access
LastValidTerminal	name of Terminal where badge was last granted access
LastValidTerminalGuid	Guid of Terminal where badge was last granted access
LastValidTime	date/time when badge was last granted access
LastInvalidBadge	badge number of Cardholder that was last denied access
LastInvalidTerminal	name of Terminal where badge was last denied access
LastInvalidTerminalGuid	Guid of Terminal where badge was last denied access
LastInvalidTime	date/time when badge was last denied access
EntryExitStatus	current status of badge (In, Out, or Undefined)

Available in P2000 V3.7 and later.

**Note:** When possible, use BadgeGetLastBadgeInfo since it is significantly more efficient.

**CardholderGetList(string sUserName, string sPassword, string sPartitionGuid, string sFirstNameFilter, string sLastNameFilter, int iMaxCount)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of partition to search (ignored on non-partitioned systems)
sFirstNameFilter	first name search filter string*
sLastNameFilter	last name search filter string*
iMaxCount	the maximum number of results to return;
* search filters can contain '*' or '%' to match any number of characters or '?' or '_' to match a single character.	

Returns a structure with the following members for each cardholder:

Member	Description
PartitionGuid	Guid of partition which owns this cardholder
PartitionName	name of partition which owns this cardholder
Public	0 = non-public, 1 = public
CardholderFirstName	Cardholder first name
CardholderMiddleName	Cardholder middle name
CardholderLastName	Cardholder last name
CardholderGuid	Cardholder Global Unique Identifier

Available in P2000 V3.6 SP1 and later.

**CardholderGetListByName(string sUserName, string sPassword, string sPartitionName, string sFirstNameFilter, string sLastNameFilter, int iMaxCount)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	name of partition to search (ignored on non-partitioned systems)
sFirstNameFilter	first name search filter string*
sLastNameFilter	last name search filter string*
iMaxCount	the maximum number of results to return;
* search filters can contain '*' or '%' to match any number of characters or '?' or '_' to match a single character.	

Returns a structure with the following members for each cardholder:

Member	Description
PartitionGuid	Guid of partition which owns this cardholder
PartitionName	name of partition which owns this cardholder
Public	0 = non-public, 1 = public
CardholderFirstName	Cardholder first name
CardholderMiddleName	Cardholder middle name
CardholderLastName	Cardholder last name
CardholderGuid	Cardholder Global Unique Identifier

Available in P2000 V3.6 SP1 and later.

**CardholderGetListCount(string sUserName, string sPassword, string sPartitionGuid, string sFirstNameFilter, string sLastNameFilter)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of partition to search (ignored on non-partitioned systems)
sFirstNameFilter	first name search filter string*
sLastNameFilter	last name search filter string*
* search filters can contain '*' or '%' to match any number of characters or '?' or '_' to match a single character.	

Returns a structure with the following member:

Member	Description
ItemValue	signed 32-bit number of matching cardholders

Available in P2000 V3.6 SP1 and later.

**CardholderGetListCountByName(string sUserName, string sPassword, string sPartitionName, string sFirstNameFilter, string sLastNameFilter)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	name of partition to search (ignored on non-partitioned systems)
sFirstNameFilter	first name search filter string*
sLastNameFilter	last name search filter string*
* search filters can contain '*' or '%' to match any number of characters or '?' or '_' to match a single character.	

Returns a structure with the following member:

Member	Description
ItemValue	signed 32-bit number of matching cardholders

Available in P2000 V3.6 SP1 and later.

**CardholderGetUdf(string sUserName, string sPassword, string sCardholderGuid, string sUdfGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sCardholderGuid	Guid of cardholder
sUdfGuid	Guid of Udf

Returns a structure with the following members for the specified Cardholder UDF:

Member	Description
ItemName	Field name
ItemValue	Value of a field

Available in P2000 V3.6 and later.

**CardholderSetEntryExitStatus(string sUserName, string sPassword, string sCardholderGuid, int iStatus)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sCardholderGuid	Guid of cardholder
iStatus	0 = undefined, 1 = in, 2 = out

Returns TRUE (1) if successful, FALSE (0) if badge does not exist, Fault Structure if failure.

Available in P2000 V3.8 SP2 and later.

**CardholderSetEntryExitStatusByID(string sUserName, string sPassword, string sEmployeeID, int iStatus)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sEmployeeID	employee ID of cardholder
iStatus	0 = undefined, 1 = in, 2 = out

Returns TRUE (1) if successful, FALSE (0) if badge does not exist, Fault Structure if failure.  
Available in P2000 V3.8 SP2 and later.

**CardholderSetUdf(string sUserName, string sPassword, string sCardholderGuid, string sUdfGuid, Value)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sCardholderGuid	Guid of cardholder
sUdfGuid	Guid of Udf
Value	UDF value of proper type for UDF (boolean, integer, string, or date)

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.6 and later.

**Company Methods****CompanyGetList(string sUserName, string sPassword, string sPartitionGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of Partition to get Company list for (ignored on non-partitioned systems)

Returns an array of structure with the following members:

Member	Description
ItemName	Company name
ItemGuid	Company Global Unique Identifier
PartitionName	name of partition which owns this Company
PartitionGuid	Guid of partition which owns this Company

Available in P2000 V3.6 SP1 and later.



**CompanyGetListByName(string sUserName, string sPassword, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	name of Partition to get Company list for (ignored on non-partitioned systems)

Returns an array of structure with the following members:

Member	Description
ItemName	Company name
ItemGuid	Company Global Unique Identifier
PartitionName	name of partition which owns this Company
PartitionGuid	Guid of partition which owns this Company

Available in P2000 V3.8 or later.

**Counter Methods****CounterGetList(string sUserName, string sPassword, string sPartitionGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of Partition to get counter list for (ignored on non-partitioned systems)

Returns a structure with the following members for the specified counter:

Member	Description
ItemName	name of counter
ItemGuid	Guid of counter
ItemValue	signed 32-bit value of counter
PartitionName	name of partition which owns this counter
PartitionGuid	Guid of partition which owns this counter
Public	0 = non-public, 1 = public

Available in P2000 V3.6 SP1 and later.

**CounterGetListByName(string sUserName, string sPassword, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	name of Partition to get counter list for (ignored on non-partitioned systems)

Returns a structure with the following members for the specified counter:

Member	Description
ItemName	name of counter
ItemGuid	Guid of counter
ItemValue	signed 32-bit value of counter

Member	Description
PartitionName	name of partition which owns this counter
PartitionGuid	Guid of partition which owns this counter
Public	0 = non-public, 1 = public

Available in P2000 V3.6 SP1 and later.

#### **CounterGetValue(string sUserName, string sPassword, string sCounterGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sCounterGuid	database GUID of counter

Returns a structure with the following member for the specified counter:

Member	Description
ItemValue	signed 32-bit value of counter

Available in P2000 V3.6 SP1 and later.

#### **CounterGetValueByName(string sUserName, string sPassword, string sCounterName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sCounterName	name of counter
sPartitionName	name of the partition (ignored on non-partitioned systems)

Returns a structure with the following member for the specified counter:

Member	Description
ItemValue	signed 32-bit value of counter

Available in P2000 V3.8 and later.

#### **CounterSetValue(string sUserName, string sPassword, string sCounterGuid, int iValue)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sCounterGuid	database GUID of counter
iValue	value to set counter to

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.6 SP1 and later.

**CounterSetValueByName(string sUserName, string sPassword, string sCounterName, string sPartitionName, int iValue)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sCounterName	name of counter
sPartitionName	name of Partition (ignored on non-partitioned systems)
iValue	value to set counter to

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**CounterChangeValue(string sUserName, string sPassword, string sCounterGuid, int iValueBy)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sCounterGuid	database GUID of counter
iValueBy	value to increment counter by

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**CounterChangeValueByName(string sUserName, string sPassword, string sCounterName, string sPartitionName, int iValueBy)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sCounterName	Counter Name
sPartitionName	Partition Name (ignored if system is not partitioned)
iValueBy	value to increment counter by

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**Department Methods****DepartmentGetList(string sUserName, string sPassword, string sPartitionGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of Partition to get Department list for (ignored on non-partitioned systems)

Returns an array of structure with the following members:

Member	Description
ItemName	Department name
ItemGuid	Department Global Unique Identifier
PartitionName	name of partition which owns this Department
PartitionGuid	Guid of partition which owns this Department

Available in P2000 V3.6 SP1 and later.

#### **DepartmentGetListByName(string sUserName, string sPassword, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	Name of Partition to get Department list for (ignored on non-partitioned systems)

Returns an array of structure with the following members:

Member	Description
ItemName	Department name
ItemGuid	Department Global Unique Identifier
PartitionName	name of partition which owns this Department
PartitionGuid	Guid of partition which owns this Department

Available in P2000 V3.8 and later.

## **Door Methods**

#### **DoorDeleteScheduled(string sUserName, string sPassword, int iScheduledID)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
iScheduledID	ID of scheduled action to be deleted

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

#### **DoorEnable(string sUserName, string sPassword, string sTerminalGuid, int iEnable)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal
iEnable	1 = enabled, 0 = disabled

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.6 and later.

**DoorGetAlarmInputs(string sUserName, string sPassword, string sTerminalGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal

Returns an array of structures with the following members:

Member	Description
AlarmName	name of alarm (Duress, Forced Door, Propped Door, Pin Code Retry, or Soft In-X-It)
InputNumber	index number of the input point
ItemName	name of input point
ItemGuid	Guid of input point
PartitionName	name of partition which owns this input point
PartitionGuid	Guid of partition which owns this input point
Public	0 = non-public, 1 = public

**Note:** If the alarm input is turned off in the P2000 configuration, all of the input parameters except AlarmName and InputNumber will be empty for that input.

Available in P2000 V3.8 and later.

**DoorGetAlarmInputsByName(string sUserName, string sPassword, string sTerminalName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalName	name of terminal
sPartitionName	name of partition (ignored for non-partitioned system)

Returns an array of structures with the following members:

Member	Description
AlarmName	name of alarm (Duress, Forced Door, Propped Door, Pin Code Retry, or Soft In-X-It)
InputNumber	index number of the input point
ItemName	name of input point
ItemGuid	Guid of input point
PartitionName	name of partition which owns this input point
PartitionGuid	Guid of partition which owns this input point
Public	0 = non-public, 1 = public

**Note:** If the alarm input is turned off in the P2000 configuration, all of the input parameters except AlarmName and InputNumber will be empty for that input.

Available in P2000 V3.8 and later.

**DoorGetList(string sUserName, string sPassword, string sPartitionGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of Partition to get door list for (ignored on non-partitioned systems)

Returns a structure with the following members for each door:

Member	Description
ItemName	name of reader terminal
ItemGuid	Guid of reader terminal
PartitionName	name of partition which owns this terminal
PartitionGuid	Guid of partition which owns this terminal
Public	0 = non-public, 1 = public
PanelName	name of panel which contains this terminal
PanelGuid	Guid of panel which contains this terminal

Available in P2000 V3.6 and later.

**DoorGetListByName(string sUserName, string sPassword, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	name of Partition to get door list for (ignored on non-partitioned systems)

Returns a structure with the following members for each door:

Member	Description
ItemName	name of reader terminal
ItemGuid	Guid of reader terminal
PartitionName	name of partition which owns this terminal
PartitionGuid	Guid of partition which owns this terminal
Public	0 = non-public, 1 = public
PanelName	name of panel which contains this terminal
PanelGuid	Guid of panel which contains this terminal

Available in P2000 V3.6 and later.

**DoorGetScheduled(string sUserName, string sPassword, string sTerminalGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal

Returns an array of structures with the following members for every action scheduled for the specified door:

Member	Description
ScheduledAction	action to be performed
ItemName	name of reader terminal
ItemGuid	Guid of reader terminal

Member	Description
ScheduledID	ID of action scheduled
ScheduledTime	time of action
ScheduledParam	override time value (only for DoorTimedOverride actions)

Available in P2000 V3.5 and later.

**DoorGetScheduledByName(string sUserName, string sPassword, string sTerminalName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalName	name of terminal
sPartitionName	name of partition (ignored for non-partitioned systems)

Returns an array of structures with the following members for every action scheduled for the specified door:

Member	Description
ScheduledAction	action to be performed
ItemName	name of reader terminal
ItemGuid	Guid of reader terminal
ScheduledID	ID of action scheduled
ScheduledTime	time of action
ScheduledParam	override time value (only for DoorTimedOverride actions)

Available in P2000 V3.8 and later.

**DoorGetAllScheduled(string sUserName, string sPassword, string sPartitionGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of Partition to get list for (ignored on non-partitioned systems)

Returns an array of structures with the following members for every action scheduled for all doors:

Member	Description
ScheduledAction	action to be performed
ItemName	name of reader terminal
ItemGuid	Guid of reader terminal
ScheduledID	ID of action scheduled
ScheduledTime	time of action
ScheduledParam	override time value (only for DoorTimedOverride actions)

Available in P2000 V3.5 and later.

**DoorGetAllScheduledByName(string sUserName, string sPassword, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	name of Partition to get list for (ignored on non-partitioned systems)

Returns an array of structures with the following members for every action scheduled for all doors:

Member	Description
ScheduledAction	action to be performed
ItemName	name of reader terminal
ItemGuid	Guid of reader terminal
ScheduledID	ID of action scheduled
ScheduledTime	time of action
ScheduledParam	override time value (only for DoorTimedOverride actions)

Available in P2000 V3.6 and later.

**DoorGetSecurityLevel(string sUserName, string sPassword, string sDoorGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sDoorGuid	Guid of door as retrieve from door point list

Returns a structure with the following members for the specified door:

Member	Description
ItemName	name of reader terminal
ItemGuid	Guid of reader terminal
ItemStatus	integer value of security level
ItemStatusText	text value of security level color ("green", "blue", "yellow", "orange", "red", or "unknown")

Available in P2000 V3.8 and later.

**DoorGetSecurityLevelByName (string sUserName, string sPassword, string sDoorName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sDoorName	name of door
sPartitionName	partition name (ignored on non-partitioned systems)

Returns a structure with the following members for the specified door:

Member	Description
ItemName	name of reader terminal
ItemGuid	Guid of reader terminal



Member	Description
ItemStatus	integer value of security level
ItemStatusText	text value of security level color ("green", "blue", "yellow", "orange", "red", or "unknown")

Available in P2000 V3.8 and later.

#### **DoorGetStatus(string sUserName, string sPassword, string sTerminalGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal

Returns a structure with the following members for the specified door:

Member	Description
ItemName	name of reader terminal
ItemGuid	Guid of reader terminal
ItemStatus	integer value of status
ItemStatusText	text value of status
ItemStatusTime	time of status in UTC

The following table lists the door status values. Please note that the applicable status values depend on the controller type.

Status Value	Status Name
0	Up / Locked
1	Down
2	Unknown
5	Unlocked
6	Disabled
7	Override
8	Lockout

Available in P2000 V3.5 and later.

#### **DoorGetStatusByName(string sUserName, string sPassword, string sTerminalName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalName	name of terminal
sPartitionName	partition name (ignored on non-partitioned systems)

Returns a structure with the following members for the specified door:

Member	Description
ItemName	name of reader terminal
ItemGuid	Guid of reader terminal
ItemStatus	integer value of status
ItemStatusText	text value of status
ItemStatusTime	time of status in UTC

The following table lists the door status values. Please note that the applicable status values depend on the controller type.

Status Value	Status Name
0	Up / Locked
1	Down
2	Unknown
5	Unlocked
6	Disabled
7	Override
8	Lockout

Available in P2000 V3.8 or later.

#### **DoorLock(string sUserName, string sPassword, string sTerminalGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

#### **DoorLockScheduled(string sUserName, string sPassword, string sTerminalGuid, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

**DoorLockByName(string sUserName, string sPassword, string sTerminalName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalName	name of terminal
sPartitionName	name of partition (ignored on non-partitioned systems)

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**DoorLockByNameScheduled(string sUserName, string sPassword, string sTerminalName, string sPartitionName, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalName	name of terminal
sPartitionName	name of partition (ignored on non-partitioned systems)
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.5 and later.

**DoorSetOverrideTimezone(string sUserName, string sPassword, string sTerminalGuid, string sTimezoneGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal
sTimezoneGuid	database GUID of the Timezone

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.5 and later.

**DoorSetOverrideTimezoneByName(string sUserName, string sPassword, string sTerminalName, string sPartitionName, string sTimezoneName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalName	name of terminal
sPartitionName	name of partition (ignored on non-partitioned systems)
sTimezoneName	name of the Timezone

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**DoorSetOverrideTimezoneEnable(string sUserName, string sPassword, string sTerminalGuid, int iEnable)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal
iEnable	1 = override timezone enabled, 0 = override timezone disabled

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.5 and later.

**DoorSetOverrideTimezoneEnableByName(string sUserName, string sPassword, string sTerminalName, string sPartitionName, int iEnable)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalName	name of terminal
sPartitionName	name of partition (ignored on non-partitioned systems)
iEnable	1 = override timezone enabled, 0 = override timezone disabled

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**DoorSetOverrideTimezoneScheduled(string sUserName, string sPassword, string sTerminalGuid, string sTimezoneGuid, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal
sTimezoneGuid	database GUID of time zone to be overridden
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.5 and later.

**DoorSetOverrideTimezoneByNameScheduled(string sUserName, string sPassword, string sTerminalName, string sPartitionName, string sTimezoneGuid, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalName	name of terminal
sPartitionName	name of partition (ignored on non-partitioned systems)
sTimezoneGuid	database GUID of time zone to be overridden
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**DoorSetOverrideTimezoneEnableScheduled(string sUserName, string sPassword, string sTerminalGuid, int iEnable, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal
iEnable	Enable flag
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.5 and later.

**DoorSetOverrideTimezoneEnableByNameScheduled (string sUserName, string sPassword, string sTerminalName, string sPartitionName, int iEnable, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalName	name of terminal
sPartitionName	name of partition (ignored on non-partitioned systems)
iEnable	Enable flag
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**DoorSetSecurityLevel(string sUserName, string sPassword, string sTerminalGuid, int iSecurityLevel)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal
iSecurityLevel	the new security level for this door (0 .. 99)

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**DoorSetSecurityLevelByName (string sUserName, string sPassword, string sTerminalName, string sPartitionName, int iSecurityLevel)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalName	name of terminal
sPartitionName	partition name (ignored on non-partitioned systems)
iSecurityLevel	the new security level for this door (0 .. 99)

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**DoorSetSecurityLevelColor(string sUserName, string sPassword, string sTerminalGuid, int iSecurityLevelColor)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal
iSecurityLevelColor	new security level for this door (0 = clear, 1 = green, 2 = blue, 3 = yellow, 4 = orange, 5 = red)

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**DoorSetSecurityLevelColorByName (string sUserName, string sPassword, string sTerminalName, string sPartitionName, string sSecurityLevelColor)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalName	name of terminal
sPartitionName	partition name (ignored on non-partitioned systems)
sSecurityLevelColor	the new security level for this door ("clear", "green", "blue", "yellow", "orange", or "red")

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**DoorTimedUnlock(string sUserName, string sPassword, string sTerminalGuid, int iTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal
iTime	amount of time to unlock door in minutes (maximum of 1440 minutes)

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.5 and later.

**DoorTimedUnlockScheduled(string sUserName, string sPassword, string sTerminalGuid, int iTime, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal
iTime	amount of time to unlock door in minutes (maximum of 1440 minutes)
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.5 and later.

**DoorTimedUnlockByName(string sUserName, string sPassword, string sTerminalName, string sPartitionName, int iTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalName	name of terminal
sPartitionName	name of partition (ignored on non-partitioned systems)
iTime	amount of time to unlock door in minutes (maximum of 1440 minutes)

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**DoorTimedUnlockByNameScheduled(string sUserName, string sPassword, string sTerminalName, string sPartitionName, int iTime, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalName	name of terminal
sPartitionName	name of partition (ignored on non-partitioned systems)
iTime	amount of time to unlock door in minutes (maximum of 1440 minutes)
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**DoorUnlock(string sUserName, string sPassword, string sTerminalGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal

Returns TRUE (1) if successful, Fault Structure if failure.  
This command will unlock the door for its programmed access time.  
Available in P2000 V3.5 and later.

**DoorUnlockScheduled(string sUserName, string sPassword, string sTerminalGuid, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalGuid	database GUID of terminal
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.  
This command will unlock the door for its programmed access time.  
Available in P2000 V3.5 and later.

**DoorUnlockByName(string sUserName, string sPassword, string sTerminalName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalName	name of terminal
sPartitionName	name of partition (ignored on non-partitioned systems)

Returns TRUE (1) if successful, Fault Structure if failure.

This command will unlock the door for its programmed access time.

Available in P2000 V3.8 and later.

**DoorUnlockByNameScheduled(string sUserName, string sPassword, string sTerminalName, string sPartitionName, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTerminalName	name of terminal
sPartitionName	name of partition (ignored on non-partitioned systems)
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.

This command will unlock the door for its programmed access time.

Available in P2000 V3.8 and later.

**DoorUnlockAll(string sUserName, string sPassword, int iUnlock, string sPartitionGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
iUnlock	0 = return to normal, 1 = unlock
sPartitionGuid	Guid of Partition of doors to unlock (ignored on non-partitioned systems)

Returns TRUE (1) if successful, Fault Structure if failure.

This command will send the “unlock all doors” command to all panels that support it.

Available in P2000 V3.5 and later.

**DoorUnlockAllByName(string sUserName, string sPassword, int iUnlock, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
iUnlock	0 = return to normal, 1 = unlock
sPartitionName	name of Partition of doors to unlock (ignored on non-partitioned systems)

Returns TRUE (1) if successful, Fault Structure if failure.

This command will send the “unlock all doors” command to all panels that support it.

Available in P2000 V3.6 and later.



**Event Methods****EventDeleteScheduled(string sUserName, string sPassword, int iScheduledID)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
iScheduledID	ID of scheduled action to be deleted

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.6 and later.

**EventGetAllScheduled(string sUserName, string sPassword, string sPartitionGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of partition to get scheduled events for (ignored for non-partitioned systems)

Returns an array of structures with the following members for every event trigger action scheduled:

Member	Description
ScheduledAction	action to be performed
ItemName	name of event to be triggered
ItemGuid	Guid of event to be triggered
PartitionName	name of partition that owns this event
PartitionGuid	Guid of partition that owns this event
Public	0 = event is non-public, 1 = event is public
ScheduledID	ID of action scheduled
ScheduledTime	time of action

Available in P2000 V3.5 and later.

**EventGetAllScheduledByName(string sUserName, string sPassword, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	name of partition to get scheduled events for (ignored for non-partitioned systems)

Returns an array of structures with the following members for every event trigger action scheduled:

Member	Description
ScheduledAction	action to be performed
ItemName	name of event to be triggered
ItemGuid	Guid of event to be triggered
PartitionName	name of partition that owns this event
PartitionGuid	Guid of partition that owns this event

Member	Description
Public	0 = event is non-public, 1 = event is public
ScheduledID	ID of action scheduled
ScheduledTime	time of action

Available in P2000 V3.6 and later.

#### **EventTrigger(string sUserName, string sPassword, string sEventGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sEventGuid	database GUID of event to be triggered

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

#### **EventTriggerByName(string sUserName, string sPassword, string sEventName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sEventName	name of event to be triggered
sPartitionName	name of partition (ignored on non-partitioned systems)

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.8 and later.

#### **EventTriggerByNameScheduled(string sUserName, string sPassword, string sEventName, string sPartitionName, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sEventName	name of event to be triggered
sPartitionName	name of partition (ignored on non-partitioned systems)
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.8 and later.

#### **EventTriggerScheduled(string sUserName, string sPassword, string sEventGuid, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sEventGuid	database GUID of event to be triggered
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

**Heart Beat Methods****HeartBeat(string sUserName, string sPassword, string sItemName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sItemName	name of the third party application

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 with SP5.

**Note:** If the HeartBeatAdd has not been called, the ItemName will be added to the list of items with a default interval value of 60 seconds.

**HeartBeatAdd(string sUserName, string sPassword, string sItemName, int iInterval)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sItemName	name of the third party application
iInterval	time in seconds between heartbeats

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.10 with SP2.

**HeartBeatDelete(string sUserName, string sPassword, string sItemName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sItemName	name of the third party application

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.10 with SP2.

## Input Methods

**InputDisable(string sUserName, string sPassword, string sInputGuid, int iDisable)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sInputGuid	database GUID of input point
iDisable	1 = disable input, 0 = enable input

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.8 and later.

**InputDisableByName(string sUserName, string sPassword, string sInputName, string sPartitionName, int iDisable)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sInputName	name of input
sPartitionName	name of partition (ignored on non-partitioned systems)
iDisable	1 = disable input, 0 = enable input

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.8 and later.

**InputGetList(string sUserName, string sPassword, string sPartitionGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of partition to get input list for (ignored on non-partitioned systems)

Returns a structure with the following members for each input:

Member	Description
ItemName	name of input
ItemGuid	Guid of input
PartitionName	name of partition which owns this input
PartitionGuid	Guid of partition which owns this input
Public	0 = non-public, 1 = public
TerminalName	name of terminal which contains this input (blank if input contained by panel)
TerminalGuid	Guid of terminal which contains this input (blank if input contained by panel)
PanelName	name of panel which contains this input
PanelGuid	Guid of panel which contains this input
SupportsSuppression	1 = suppression supported, 0 = suppression not supported

Available in P2000 V3.6 and later.

**InputGetListByName(string sUserName, string sPassword, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	name of partition to get Input list for (ignored on non-partitioned systems)

Returns a structure with the following members for each input:

Member	Description
ItemName	name of input
ItemGuid	Guid of input
PartitionName	name of partition which owns this input
PartitionGuid	Guid of partition which owns this input
Public	0 = non-public, 1 = public
TerminalName	name of terminal which contains this input (blank if input contained by panel)
TerminalGuid	Guid of terminal which contains this input (blank if input contained by panel)
PanelName	name of panel which contains this input
PanelGuid	Guid of panel which contains this input
SupportsSuppression	1 = suppression supported, 0 = suppression not supported

Available in P2000 V3.6 and later.

**InputGetStatus(string sUserName, string sPassword, string sInputGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sInputGuid	database GUID of input point

Returns a structure with the following members for the specified input:

Member	Description
ItemName	name of input
ItemGuid	Guid of input
ItemStatus	integer value of status
ItemStatusText	text value of status
ItemStatusTime	time of status in UTC

**Input Status Values**

INPUT_STATUS_DISABLED	-1	"Disabled"
INPUT_STATUS_SECURE	0	"Secure"
INPUT_STATUS_SET	1	"Set"
INPUT_STATUS_SHORT	2	"Short"
INPUT_STATUS_OPEN	3	"Open"
INPUT_STATUS_SUPPRESSED	4	"Suppressed"
INPUT_STATUS_UNKNOWN	5	"Unknown"

Available in P2000 V3.5 and later.

**InputGetStatusByName(string sUserName, string sPassword, string sInputName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sInputName	name of input
sPartitionName	name of partition (ignored on non-partitioned systems)

Returns a structure with the following members for the specified input:

Member	Description
ItemName	name of input
ItemGuid	Guid of input
ItemStatus	integer value of status
ItemStatusText	text value of status
ItemStatusTime	time of status in UTC

**Input Status Values**

INPUT_STATUS_DISABLED	-1	"Disabled"
INPUT_STATUS_SECURE	0	"Secure"
INPUT_STATUS_SET	1	"Set"
INPUT_STATUS_SHORT	2	"Short"
INPUT_STATUS_OPEN	3	"Open"
INPUT_STATUS_SUPPRESSED	4	"Suppressed"
INPUT_STATUS_UNKNOWN	5	"Unknown"

Available in P2000 V3.8 and later.

**InputSuppress(string sUserName, string sPassword, string sInputGuid, int iDuration)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sInputGuid	database GUID of input point
iDuration	duration in seconds (0 to cancel suppression)

Returns TRUE (1) if successful, FALSE (0) if not supported, Fault Structure if failure.

Available in P2000 V3.8 and later.

**InputSuppressByName(string sUserName, string sPassword, string sInputName, string sPartition, int iDuration)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sInputName	name of input
sPartition	name of partition (ignored on non-partitioned systems)
iDuration	duration in seconds (0 to cancel suppression)

Returns TRUE (1) if successful, FALSE (0) if not supported, Fault Structure if failure.

Available in P2000 V3.8 and later.

## Output Methods

Some of the following methods refer to Output Point Actions. Not all output point actions are supported by all panel types. The following table shows the different output actions:

Action	Name
0	Reset
1	Set
2	Slow Flash
3	Fast Flash
4	Preset

### OutputDeleteScheduled(string sUserName, string sPassword, int iScheduledID)

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
iScheduledID	ID of scheduled action to be deleted

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.5 and later.

### OutputGetList(string sUserName, string sPassword, string sPartitionGuid)

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of Partition to get Output list for (ignored on non-partitioned systems)

Returns a structure with the following members for each output:

Member	Description
ItemName	name of output
ItemGuid	Guid of output
PartitionName	name of partition which owns this output
PartitionGuid	Guid of partition which owns this output
Public	0 = non-public, 1 = public
TerminalName	name of terminal which contains this output
TerminalGuid	Guid of terminal which contains this output
PanelName	name of panel which contains this output
PanelGuid	Guid of panel which contains this output

Available in P2000 V3.6 and later.

**OutputGetListByName(string sUserName, string sPassword, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	name of partition to get Output list for (ignored on non-partitioned systems)

Returns a structure with the following members for each output:

Member	Description
ItemName	name of output
ItemGuid	Guid of output
PartitionName	name of partition which owns this output
PartitionGuid	Guid of partition which owns this output
Public	0 = non-public, 1 = public
TerminalName	name of terminal which contains this output
TerminalGuid	Guid of terminal which contains this output
PanelName	name of panel which contains this output
PanelGuid	Guid of panel which contains this output

Available in P2000 V3.6 and later.

**OutputGetScheduled(string sUserName, string sPassword, string sOutputGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sOutputGuid	database GUID of output point

Returns an array of structures with the following members for every action scheduled for the specified output point:

Member	Description
ScheduledAction	action to be performed
ItemName	name of output point
ItemGuid	Guid of output point
ScheduledID	ID of action scheduled
ScheduledTime	time of action
ScheduledParam	duration

Available in P2000 V3.5 and later.



**OutputGetScheduledByName(string sUserName, string sPassword, string sOutputName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sOutputName	name of output point
sPartitionName	name of partition (ignored on non-partitioned systems)

Returns an array of structures with the following members for every action scheduled for the specified output point:

Member	Description
ScheduledAction	action to be performed
ItemName	name of output point
ItemGuid	Guid of output point
ScheduledID	ID of action scheduled
ScheduledTime	time of action
ScheduledParam	duration

Available in P2000 V3.8 and later.

**OutputGetStatus(string sUserName, string sPassword, string sOutputGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sOutputGuid	database GUID of output point

Returns a structure with the following members for the specified output:

Member	Description
ItemName	name of output
ItemGuid	Guid of output
ItemStatus	integer value of status
ItemStatusText	text value of status
ItemStatusTime	time of status in UTC

The possible status values for an output point are listed in the following table.

Status	Name
0	Reset
1	Set
2	Unknown

Available in P2000 V3.5 and later.

**OutputGetStatusByName(string sUserName, string sPassword, string sOutputName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sOutputName	name of output
sPartitionName	name of partition (ignored on non-partitioned systems)

Returns a structure with the following members for the specified output:

Member	Description
ItemName	name of output
ItemGuid	Guid of output
ItemStatus	integer value of status
ItemStatusText	text value of status
ItemStatusTime	time of status in UTC

The possible status values for an output point are listed in the following table:

Status	Name
0	Reset
1	Set
2	Unknown

Available in P2000 V3.8 and later.

**OutputSet(string sUserName, string sPassword, string sOutputGuid, int iAction)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sOutputGuid	database GUID of output point
iAction	0 = reset output, 1 = set output, 2 = slow flash, 3 = fast flash, 4 = pre-set

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

**OutputSetByName(string sUserName, string sPassword, string sOutputName, string sPartitionName, int iAction)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sOutputName	name of output point
sPartitionName	name of partition (ignored on non-partitioned systems)
iAction	0 = reset output, 1 = set output, 2 = slow flash, 3 = fast flash, 4 = pre-set

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.8 and later.

**OutputSetByNameScheduled(string sUserName, string sPassword, string sOutputName, string sPartitionName, int iAction, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sOutputName	name of output point
sPartitionName	name of partition (ignored on non-partitioned systems)
iAction	0 = reset output, 1 = set output
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**OutputSetForDuration(string sUserName, string sPassword, string sOutputGuid, int iAction, int iDuration)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sOutputGuid	database GUID of output point
iAction	0 = reset output, 1 = set output
iDuration	duration in seconds (ignored if action is reset)

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.5 and later.

**OutputSetForDurationByName(string sUserName, string sPassword, string sOutputName, string sPartitionName, int iAction, int iDuration)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sOutputName	name of output point
sPartitionName	name of partition (ignored on non-partitioned systems)
iAction	0 = reset output, 1 = set output
iDuration	duration in seconds (ignored if action is reset)

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**OutputSetForDurationScheduled(string sUserName, string sPassword, string sOutputGuid, int iAction, int iDuration, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sOutputGuid	database GUID of output point
iAction	0 = reset output, 1 = set output
iDuration	duration in seconds (ignored if action is reset)
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

**OutputSetForDurationByNameScheduled(string sUserName, string sPassword, string sOutputName, string sPartitionName, int iAction, int iDuration, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sOutputName	name of output point
sPartitionName	name of partition (ignored on non-partitioned systems)
iAction	0 = reset output, 1 = set output
iDuration	duration in seconds (ignored if action is reset)
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

**OutputSetScheduled(string sUserName, string sPassword, string sOutputGuid, int iAction, datetime tScheduledTime)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sOutputGuid	database GUID of output point
iAction	0 = reset output, 1 = set output
tScheduledTime	time for action to occur

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

## P2000 Methods

### P2000GetVersion(string sUserName, string sPassword)

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password

Returns a structure with the following members:

Member	Description
MajorVersion	major version number
MinorVersion	minor version number

Available in P2000 V3.8 and later.

## Panel Methods

### PanelGetComponents(string sUserName, string sPassword, string sPanelGuid, string sPartitionGuid)

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPanelGuid	Guid of panel
sPartitionGuid	Guid of Partition to get component list for (ignored on non-partitioned systems)

Returns a structure array with sections for Terminals, Inputs, and Outputs.

Members for each Terminal	Description
ItemName	name of reader terminal
ItemGuid	Guid of reader terminal
PartitionName	name of partition which owns this terminal
PartitionGuid	Guid of partition which owns this terminal
Public	0 = non-public, 1 = public
InputTerminal	0 = not an input terminal, 1 = input terminal
OutputTerminal	0 = not an output terminal, 1 = output terminal
ReaderTerminal	0 = not a reader terminal, 1 = reader terminal
PanelName	name of panel which contains this terminal
PanelGuid	Guid of panel which contains this terminal

Members for each Input	Description
ItemName	name of input
ItemGuid	Guid of input
PartitionName	name of partition which owns this input
PartitionGuid	Guid of partition which owns this input
Public	0 = non-public, 1 = public
TerminalName	name of terminal which contains this input (blank if input contained by panel)

Members for each Input	Description
TerminalGuid	Guid of terminal which contains this input (blank if input contained by panel)
PanelName	name of panel which contains this input
PanelGuid	Guid of panel which contains this input
SupportsSuppression	1 = suppression supported, 0 = suppression not supported

Members for each Output	Description
ItemName	name of output
ItemGuid	Guid of output
PartitionName	name of partition which owns this output
PartitionGuid	Guid of partition which owns this output
Public	0 = non-public, 1 = public
TerminalName	name of terminal which contains this output
TerminalGuid	Guid of terminal which contains this output
PanelName	name of panel which contains this output
PanelGuid	Guid of panel which contains this output

Available in P2000 V3.8 and later.

**PanelGetComponentsByName(string sUserName, string sPassword, string sPanelName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPanelName	name of panel
sPartitionName	name of Partition to get component list for (ignored on non-partitioned systems)

Returns a structure array with sections for Terminals, Inputs, and Outputs.

Members for each Terminal	Description
ItemName	name of reader terminal
ItemGuid	Guid of reader terminal
PartitionName	name of partition which owns this terminal
PartitionGuid	Guid of partition which owns this terminal
Public	0 = non-public, 1 = public
InputTerminal	0 = not an input terminal, 1 = input terminal
OutputTerminal	0 = not an output terminal, 1 = output terminal
ReaderTerminal	0 = not an reader terminal, 1 = reader terminal
PanelName	name of panel which contains this terminal
PanelGuid	Guid of panel which contains this terminal

Members for each Input	Description
ItemName	name of input
ItemGuid	Guid of input
PartitionName	name of partition which owns this input
PartitionGuid	Guid of partition which owns this input
Public	0 = non-public, 1 = public

Members for each Input	Description
TerminalName	name of terminal which contains this input (blank if input contained by panel)
TerminalGuid	Guid of terminal which contains this input (blank if input contained by panel)
PanelName	name of panel which contains this input
PanelGuid	Guid of panel which contains this input
SupportsSuppression	1 = suppression supported, 0 = suppression not supported

Members for each Output	Description
ItemName	name of output
ItemGuid	Guid of output
PartitionName	name of partition which owns this output
PartitionGuid	Guid of partition which owns this output
Public	0 = non-public, 1 = public
TerminalName	name of terminal which contains this output
TerminalGuid	Guid of terminal which contains this output
PanelName	name of panel which contains this output
PanelGuid	Guid of panel which contains this output

Available in P2000 V3.8 and later.

#### PanelGetList(string sUserName, string sPassword, string sPartitionGuid)

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of Partition to get Panel list for (ignored on non-partitioned systems)

Returns a structure with the following members for each panel:

Member	Description
ItemName	name of panel
ItemGuid	Guid of panel
ItemModel	model name of panel
PartitionName	name of partition which owns this panel
PartitionGuid	Guid of partition which owns this panel
Public	0 = non-public, 1 = public

Available in P2000 V3.6 and later.

**PanelGetListByName(string sUserName, string sPassword, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	name of Partition to get Panel list for (ignored on non-partitioned systems)

Returns a structure with the following members for each panel:

Member	Description
ItemName	name of panel
ItemGuid	Guid of panel
ItemModel	model name of panel
PartitionName	name of partition which owns this panel
PartitionGuid	Guid of partition which owns this panel
Public	0 = non-public, 1 = public

Available in P2000 V3.6 and later.

**PanelGetTimezones(string sUserName, string sPassword, string sPanelGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPanelGuid	Guid of panel

Returns an array of structures with the following members for every timezone assigned to the specified panel:

Member	Description
TimezonePosition	position of timezone
TimezoneName	name of timezone ("<NULL>" if none)
TimezoneGuid	Guid of Timezone ("<NULL>" if none)
OutputGroupName	name of output group ("<NULL>" if none)
OutputGroupGuid	Guid of output group ("<NULL>" if none)

Available in P2000 V3.5 and later.

**PanelGetTimezonesByName(string sUserName, string sPassword, string sPanelName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPanelName	name of panel
sPartitionName	name of partition (ignored on non-partitioned systems)

Returns an array of structures with the following members for every timezone assigned to the specified panel:

Member	Description
TimezonePosition	position of timezone
TimezoneName	name of timezone ("<NULL>" if none)



Member	Description
TimezoneGuid	Guid of Timezone ("<NULL>" if none)
OutputGroupName	name of output group ("<NULL>" if none)
OutputGroupGuid	Guid of output group ("<NULL>" if none)

Available in P2000 V3.5 and later.

**PanelModifyTimezone(string sUserName, string sPassword, string sPanelGuid, int iTimezonePos, string sTimezoneGuid, string sOutputGrpGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPanelGuid	Guid of panel
iTimezonePos	position of Timezone (1 to 64)
sTimezoneGuid	Guid of Timezone ("<NULL>" if none)
sOutputGrpGuid	Guid of output group ("<NULL>" if none)

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

**PanelModifyTimezoneByName(string sUserName, string sPassword, string sPanelName, string sPartitionName, int iTimezonePos, string sTimezoneName, string sOutputGrpName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPanelName	name of panel
sPartitionName	name of partition (ignored on non-partitioned systems)
iTimezonePos	position of Timezone (1 to 64)
sTimezoneName	name of Timezone ("<NULL>" if none)
sOutputGrpName	name of output group ("<NULL>" if none)

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

## Partition Methods

**PartitionGetList(string sUserName, string sPassword)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password

Returns an array of structures with the following members for each partition:

Member	Description
PartitionName	name of partition
PartitionGuid	Guid of partition

Available in P2000 V3.6 and later.

## Terminal Methods

### TerminalGetList(string sUserName, string sPassword, string sPartitionGuid)

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of Partition to get terminal list for (ignored on non-partitioned systems)

Returns a structure with the following members for each terminal:

Member	Description
ItemName	name of reader terminal
ItemGuid	Guid of reader terminal
PartitionName	name of partition which owns this terminal
PartitionGuid	Guid of partition which owns this terminal
Public	0 = non-public, 1 = public
InputTerminal	0 = not an input terminal, 1 = input terminal
OutputTerminal	0 = not an output terminal, 1 = output terminal
ReaderTerminal	0 = not a reader terminal, 1 = reader terminal
PanelName	name of panel which contains this terminal
PanelGuid	Guid of panel which contains this terminal

Available in P2000 V3.6 and later.

### TerminalGetListByName(string sUserName, string sPassword, string sPartitionName)

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	name of Partition to get terminal list for (ignored on non-partitioned systems)

Returns a structure with the following members for each terminal:

Member	Description
ItemName	name of reader terminal
ItemGuid	Guid of reader terminal
PartitionName	name of partition which owns this terminal
PartitionGuid	Guid of partition which owns this terminal
Public	0 = non-public, 1 = public
InputTerminal	0 = not an input terminal, 1 = input terminal
OutputTerminal	0 = not an output terminal, 1 = output terminal
ReaderTerminal	0 = not a reader terminal, 1 = reader terminal
PanelName	name of panel which contains this terminal
PanelGuid	Guid of panel which contains this terminal

Available in P2000 V3.6 and later.

## Timezone Methods

**TimezoneAdd(string sUserName, string sPassword, string sTimezoneName, string sTimezoneGuid, string sPartitionGuid, int iPublic, int iAddToAllPanels, int array iInitialStates, datetime array tTimes)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTimezoneName	name for new Timezone
sTimezoneGuid	GUID for new Timezone (autoassigned if "NULL")
sPartitionGuid	GUID for the partition which owns this timezone (ignored on non-partitioned systems)
iPublic	0 = non-public, 1 = public (ignored on non-partitioned systems)
iAddToAllPanels	0 = do not add to panels, 1 = add to all panels
iInitialStates	array of 10 initial states (0 = starts disabled at midnight, 1 = starts enabled at midnight)
tTimes	array of 70 timezone toggles (ignored if 00:00:00)

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

**Note:** This method does not support setting elevator floor groups settings of timezones.

**TimezoneAddByName(string sUserName, string sPassword, string sTimezoneName, string sTimezoneGuid, string sPartitionName, int iPublic, int iAddToAllPanels, int array iInitialStates, datetime array tTimes)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTimezoneName	name for new Timezone
sTimezoneGuid	GUID for new Timezone (autoassigned if "NULL")
sPartitionName	name of the partition which owns this timezone (ignored on non-partitioned systems)
iPublic	0 = non-public, 1 = public (ignored on non-partitioned systems)
iAddToAllPanels	0 = do not add to panels, 1 = add to all panels
iInitialStates	array of 10 initial states (0 = starts disabled at midnight, 1 = starts enabled at midnight)
tTimes	array of 70 timezone toggles (ignored if 00:00:00)

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

**Note:** This method does not support setting elevator floor group settings of timezones.

**TimezonelsActive(string sUserName, string sPassword, string sTimezoneGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTimezoneGuid	database GUID of the Timezone

Returns TRUE (1) if Timezone is active, FALSE (0) if inactive, Fault Structure if failure.  
Available in P2000 V3.6 SP1 and later.

**TimezonelsActiveByName(string sUserName, string sPassword, string sTimezoneName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTimezoneName	name of the Timezone
sPartitionName	name of partition (ignored on non-partitioned systems)

Returns TRUE (1) if Timezone is active, FALSE (0) if inactive, Fault Structure if failure.  
Available in P2000 V3.8 and later.

**TimezoneGetInfo(string sUserName, string sPassword, string sTimezoneGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTimezoneGuid	database GUID of the Timezone

Returns a structure with the following members for the specified timezone:

Member	Description
ItemName	Timezone name
ItemGuid	Timezone Global Unique Identifier
PartitionGuid	Guid of partition which owns this timezone
PartitionName	name of partition which owns this timezone
Public	0 = non-public, 1 = public
InitialStates	an array of 10 initial states (0 = disabled at midnight, 1 = enabled at midnight)
ToggleTimes	an array of 70 timezone toggles (unused if 00:00:00)

Available in P2000 V3.6 SP1 and later.

**TimezoneGetInfoByName(string sUserName, string sPassword, string sTimezoneName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTimezoneName	name of the Timezone
sPartitionName	name of partition (ignored on non-partitioned systems)

Returns a structure with the following members for the specified timezone:

Member	Description
ItemName	Timezone name
ItemGuid	Timezone Global Unique Identifier
PartitionGuid	Guid of partition which owns this timezone
PartitionName	name of partition which owns this timezone
Public	0 = non-public, 1 = public
InitialStates	an array of 10 initial states (0 = disabled at midnight, 1 = enabled at midnight)
ToggleTimes	an array of 70 timezone toggles (unused if 00:00:00)

Available in P2000 V3.8 and later.

**TimezoneGetList(string sUserName, string sPassword, string sPartitionGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of Partition to get Timezone list for (ignored on non-partitioned systems)

Returns an array of structures with the following members for the each timezone:

Member	Description
ItemName	Timezone name
ItemGuid	Timezone Global Unique Identifier
PartitionGuid	Guid of partition which owns this timezone
PartitionName	name of partition which owns this timezone
Public	0 = non-public, 1 = public

Available in P2000 V3.6 SP1 and later.

**TimezoneGetListByName(string sUserName, string sPassword, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	name of Partition to get Timezone list for (ignored on non-partitioned systems)

Returns an array of structures with the following members for the each timezone:

Member	Description
ItemName	Timezone name
ItemGuid	Timezone Global Unique Identifier
PartitionGuid	Guid of partition which owns this timezone
PartitionName	name of partition which owns this timezone
Public	0 = non-public, 1 = public

Available in P2000 V3.6 SP1 and later.

**TimezoneModify(string sUserName, string sPassword, string sTimezoneGuid, int  
iInitialState, datetime array tTimes)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTimezoneGuid	database GUID of the Timezone
iInitialStates	array of 10 initial states (0 = starts disabled at midnight, 1 = starts enabled at midnight)
tTimes	array of 70 timezone toggles (ignored if 00:00:00)

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

**Note:** This method does not support setting elevator floor groups settings of timezones.

**TimezoneModifyByName(string sUserName, string sPassword, string sTimezoneName,  
string sPartitionName, int iInitialState, datetime array tTimes)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTimezoneName	name of the Timezone
sPartitionName	name of Partition (ignored on non-partitioned systems)
iInitialStates	array of 10 initial states (0 = starts disabled at midnight, 1 = starts enabled at midnight)
tTimes	array of 70 timezone toggles (ignored if 00:00:00)

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

**Note:** This method does not support setting elevator floor groups settings of timezones.

**TimezoneDelete(string sUserName, string sPassword, string sTimezoneGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTimezoneGuid	database GUID of the Timezone

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.5 and later.

**Note:** The timezone will be deleted from all panels and all badges that are assigned this timezone.

**TimezoneDeleteByName(string sUserName, string sPassword, string sTimezoneName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sTimezoneName	name of the Timezone
sPartitionName	name of partition (ignored if system is not partitioned)

Returns TRUE (1) if successful, Fault Structure if failure.  
Available in P2000 V3.5 and later.

**Note:** The timezone will be deleted from all panels and all badges that are assigned this timezone.

**UDF Methods****UdfGetList(string sUserName, string sPassword, string sPartitionGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionGuid	Guid of Partition to get UDF list for (ignored on non-partitioned systems)

Returns an array of structure with the following members:

Member	Description
ItemName	UDF field name
ItemType	Data type of UDF field
ItemGuid	Guid of UDF
ItemWidth	width of UDF field
ItemOrder	order of UDF field
PartitionName	name of partition which owns this UDF
PartitionGuid	Guid of partition which owns this UDF
Public	0 = non-public, 1 = public

The following table lists the UDF data type values:

Type Value	Type Name
0	Text
1	Numeric
2	Boolean
3	Date
4	Selection

Available in P2000 V3.6 and later.

**UdfGetListByName(string sUserName, string sPassword, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sPartitionName	name of Partition to get UDF list for (ignored on non-partitioned systems)

Returns an array of structure with the following members:

Member	Description
ItemName	UDF field name
ItemType	Data type of UDF field
ItemGuid	Guid of UDF field
ItemWidth	width of UDF field
ItemOrder	order of UDF field
PartitionName	name of partition which owns this UDF
PartitionGuid	Guid of partition which owns this UDF
Public	0 = non-public, 1 = public

The following table lists the UDF data type values:

Type Value	Type Name
0	Text
1	Numeric
2	Boolean
3	Date
4	Selection

Available in P2000 V3.6 and later.

**UdfGetSelectionChoices(string sUserName, string sPassword, string sUdfGuid)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sUdfGuid	Guid of UDF to get selection choices for

Returns an array of structure with the following member:

Member	Description
ItemName	Selection choice

Available in P2000 V3.9 and later.



**UdfGetSelectionChoicesByName(string sUserName, string sPassword, string sUdfName, string sPartitionName)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
sUdfName	name of UDF to get selection choices for
sPartitionName	name of partition (ignored on non-partitioned systems)

Returns an array of structure with the following member:

Member	Description
ItemName	Selection choice

Available in P2000 V3.9 and later.

**XAction Methods****XActionSave(string sUserName, string sPassword, int iHistType, string sPanelName, string sTerminalName, string sPointName, string sBadgeNumber, string sFirstName, string sLastName, int iTimedOverride, int iIssueLevel, int iFacilityCode, string sEventName, datetime tTime, string sPartition, int iPublic)**

Argument	Description
sUserName	valid P2000 username
sPassword	valid P2000 password
iHistType	History Type
sPanelName	name of panel
sTerminalName	Terminal name
sPointName	Point name
sBadgeNumber	Badge number
sFirstName	First name
sLastName	Last name
iTimedOverride	Override time
iIssueLevel	Issue Level of badge
iFacilityCode	Facility code of badge
sEventName	name of event
tTime	timestamp of transaction
sPartition	name of partition which owns this action.
iPublic	0 = non-public, 1 = public

Returns TRUE (1) if successful, Fault Structure if failure.

Available in P2000 V3.6 and later.

PRELIMINARY