November 12, 2021

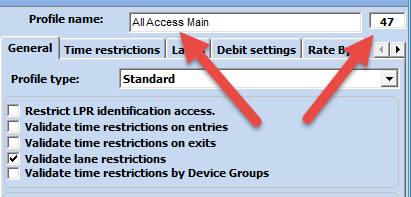
**Statement of Work – OpenPath to TIBA SmartPark Interface**

The purpose of this document is to define the statement of work (SOW) for an integration between OpenPath and TIBA SmartPark. OpenPath is an access control system utilized in multiple industries. OpenPath is a reliable mobile access control with scalable cloud-based software and endless integration capabilities. Fast. Flexible. Future-proof.

TIBA and OpenPath desire to create an interface whereby the users of OpenPath access control system can utilize their credentials to access the TIBA SmartPark PARCS system. OpenPath will be the “master” in maintaining the users and credentials. Custom fields within OpenPath will be used to allow the specification of the company, sub-company, parking privileges and cell phone number to be used in SmartPark.

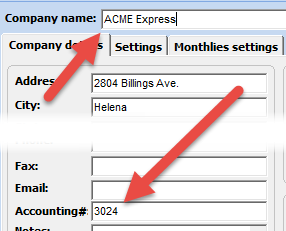
OpenPath has an API that can be utilize to retrieve user information for adding or updating users in SmartPark. The interface will be defined such that it can be scheduled to periodically look for new or updated users in OpenPath to add or update monthly parkers in SmartPark.

Statement of Work

**Setup Items** – Coordination will be needed between an OpenPath site and the SmartPark PARCS system.

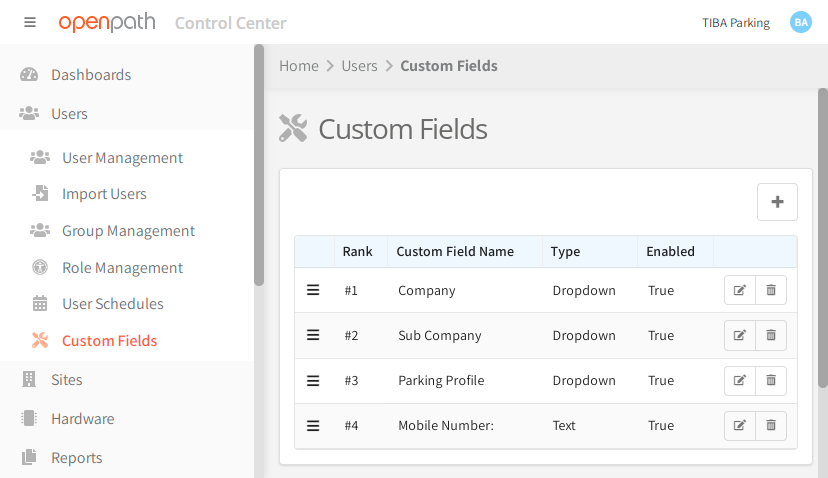
In SmartPark: the parking access items (Access Profiles) will need to be defined with unique numbers. The Access Profile numbers and description recorded for use in OpenPath.

Additionally, all Companies and Sub Companies will be defined in SmartPark with a unique Accounting # (Company ID) recorded for use in OpenPath.

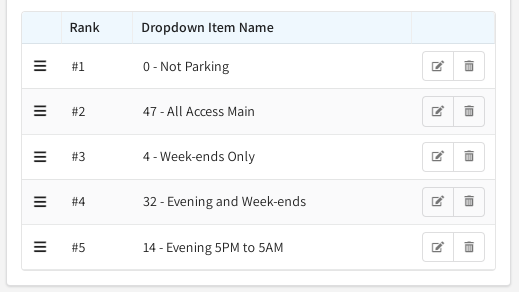


In OpenPath: the recorded Access profiles, Companies and Sub companies from SmartPark will be used to build values for several custom fields in OpenPath.

<https://control.openpath.com> user: bernie@amc-hln.com pwd: !Dakota23op ORG ID: 5995

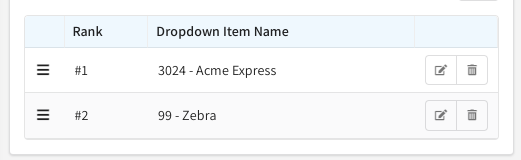
Suggested custom fields to be defined in OpenPath are:  


The values from the SmartPark access profiles and Company lists will be used to populate the values in the custom fields. For example, the access profiles in the OpenPath “Parking Profile” custom field:

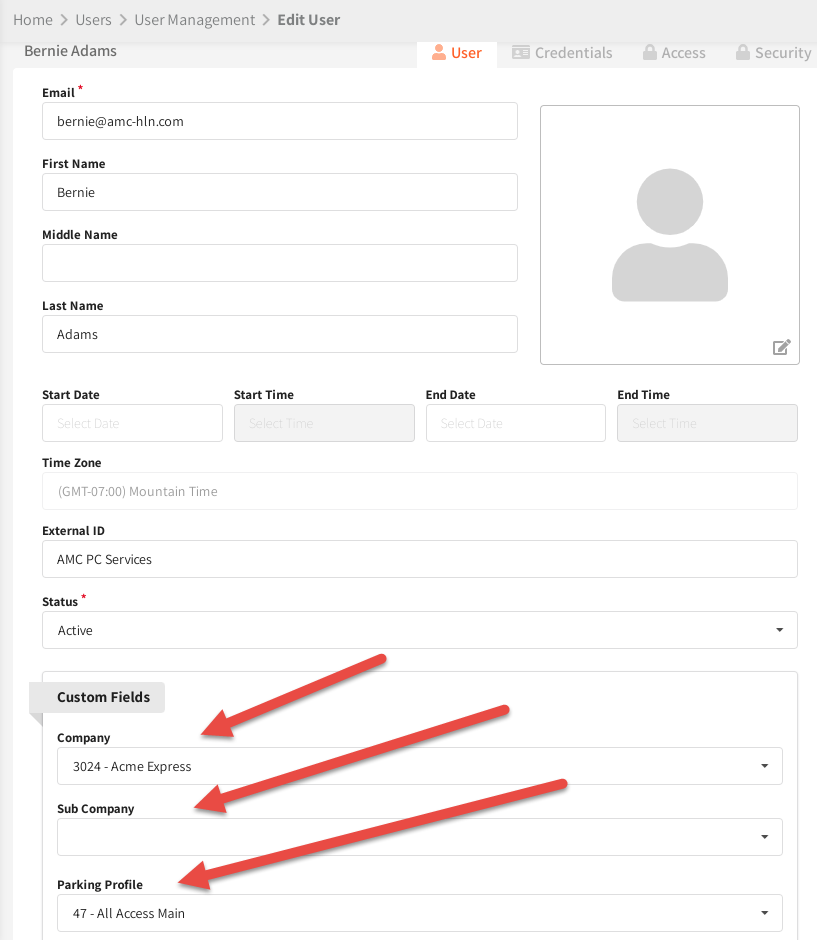
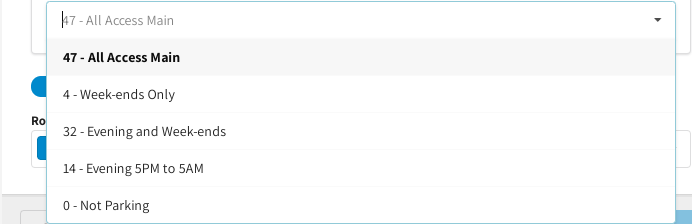


Note the custom field dropdown description will begin with the SmartPark Access profile number followed by a dash and then the description. This is important as the Access profile number is a critical value for adding and updating monthlies in SmartPark.

For Companies and Sub Companies, the same pattern should be used. For example:



The defined dropdown lists will be presented to the OpenPath operator when adding or updating users. The operator would utilize the drop-down to select the pre-defined value to be saved with that user.

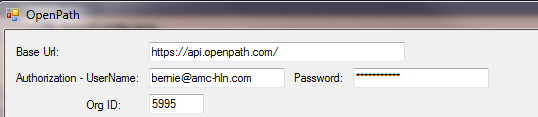


Setup items for the interface include the selection of a default access profile number and company ID in case the custom field value specified in the OpenPath custom field does not match an existing Access profile number, Company ID or Sub-Company ID.

The above setup items are in addition to the required parameters for the OpenPath API, SmartPark vendor API and SmartPark SQL Server connectivity.

 In SmartPark, an external vendor account will be defined for usage of the API (Tools/Tables/SmartConnect Vendor Accounts). AMC has been assigned vendor ID 57. Once the external vendor account is set up, the values will need to be provided for use in the TTSP program. The necessary values are: Facility ID, Vendor ID, Terminal number, user ID, password.

Additionally, the program will require the ability to connect to the TIBA SQL server for creating and using a custom database. Access to the TBPark database will be via API calls. Values need are: SQL Server instance address, SQL user name and SQL password.

For OpenPath, the required parameters will be: OpenPath URL, user name/email, password and Org ID#.

Setup will also require retrieving the ID number from OpenPath for each of the custom fields defined in OpenPath. Specifically, the Custom field ID for Companies, Sub Companies, Access Profiles and cell phone.

Additionally, the values defined for each of the custom fields retrieved will be saved for easier look-up during processing.

There will be other setup items that will become apparent as the system is developed.

**Expected Programs, Processes and Databases** – The general list of the required programs and processes will be listed and their function.

Over All Menu – a menu will be available to make it easier to access the various processes and programs.

A custom database named AMC\_SRP\_DATABASE may be defined on the TIBASQL SQL server. The database will contain the setup items including the list of the values for each of the custom tables. There may be additional data or tables defined as needed to support the application.

OpenPath Framework – process defined to verify OpenPath API operations, facilitate the proper parsing of data, standardize interactions and troubleshoot operation. Processes, modules and procedures defined in the framework will be utilized in any or all of the other programs or processes that require the features.

The OpenPath API utilizes a RESTful web interface and returns json formatted responses. Each command utilized will require authentication, query items or parameters for the command. The view the OpenPath API, proceed to this web address: <https://openpath.readme.io/> <https://openpath.readme.io/reference/> The web client calls will generally look as follows: “https://api.openpath.com/orgs/9999/command”. The OpenPath API commands that will be utilized for this interface are: ‘zones’ (list all zones), ‘customFields’ (list all custom fields), ‘users’ (list of users), ‘users/9999999/credentials’ (list all credentials for specified user). The “zones” command will generally be used to verify connectivity to the OpenPath API.

All processes will log their actions. If any critical error occurs related to data received from OpenPath (I.E. Company ID not found or Access Profile ID not found) will be recorded in a separate file. This allows the errors to be reviewed and corrections made to either SmartPark or to OpenPath. Each program or process will be responsible for purging older log and error files (older than 60 days).

**General Setup** – Process to define the “static” setup items and to map custom fields to their purposes (Company, Sub Company, Access Profile or cell phone).

Custom Fields – the OpenPath API command “customfields” will return a list of the defined custom fields and the assigned values for each. Data will be used to update the AMC\_SRP\_DATABASE. A sample of the returned data from the command is as follows:

{

"data": [

{

"id": 283,

"name": "Company",

"ordinal": 1,

"isEnabled": true,

"createdAt": "2021-09-14T14:41:59.000Z",

"updatedAt": "2021-09-14T14:50:00.000Z",

"customFieldType": {

"id": 4,

"name": "Dropdown"

},

"customFieldDropdownItems": [

{

"id": 302,

"name": "3024 - Acme Express"

},

{

"id": 303,

"name": "99 - Zebra"

}

]

},

{

"id": 284,

"name": "Sub Company",

"ordinal": 2,

"isEnabled": true,

"createdAt": "2021-09-14T14:46:06.000Z",

"updatedAt": "2021-09-14T14:50:00.000Z",

"customFieldType": {

"id": 4,

"name": "Dropdown"

},

"customFieldDropdownItems": [

{

"id": 304,

"name": "13024 - Acme Express - Accounting"

},

{

"id": 306,

"name": "4 - ACME Express - Sales"

}

]

},

{

"id": 282,

"name": "Parking Profile",

"ordinal": 3,

"isEnabled": true,

"createdAt": "2021-09-14T14:32:54.000Z",

"updatedAt": "2021-09-14T14:50:00.000Z",

"customFieldType": {

"id": 4,

"name": "Dropdown"

},

"customFieldDropdownItems": [

{

"id": 298,

"name": "47 - All Access Main"

},

{

"id": 299,

"name": "4 - Week-ends Only"

},

{

"id": 300,

"name": "32 - Evening and Week-ends"

},

{

"id": 301,

"name": "14 - Evening 5PM to 5AM"

},

{

"id": 305,

"name": "0 - Not Parking"

}

]

}

],

"meta": {

"accessToken": {

"scopeUpdatedAt": "2021-09-14T14:31:47.000Z"

},

"siteSpecificAccess": null

},

"totalCount": 3,

"filteredCount": 3

}

The id: value for each custom field is unique for that custom field. I.E.

{

"id": 283,

"name": "Company"

The id value “283” denotes the value for the overall Company custom field. This is the identifier that will be used to map the OpenPath custom field to the “Company” for SmartPark.

In the customFieldDropdownItems, the individual selection for the Company the user works are listed.

"customFieldDropdownItems": [

{

"id": 302,

"name": "3024 - Acme Express"

},

{

"id": 303,

"name": "99 - Zebra"

}

]

When the user information is retrieved from OpenPath (see example later in this document), the custom fields section of the user will contain the overall custom field ID and name (I.E. 283 Company) and the selected value (I.E. 302 denoting 3024 - Acme Express).

The interface will map the various custom field ids to their uses. Companies, Sub-Companies, Access Profiles, Cell Phone. Optionally, for those custom fields that have dropdown list items that should tie to SmartPark, the program will verify that the values listed are valid within SmartPark.

As stated previously, there may be other setup items that become apparent as the application is developed.

**Changes Processing Overview**

Changes processing (CP) can be either run on-demand or as a scheduled task to run as frequently as desired. CP will first verify it can communicate to the SQL server hosting the AMC\_SRP\_DATABASE, OpenPath and to SmartPark API. Next, the program will check the customFields to determine if any new dropdown items have been added or removed. Process may also verify the dropdown values against SmartPark system. For speed purposes, the Companies and Access Profiles in SmartPark will be pre-loaded to a temporary table for use in the interface to validate data.

CP will proceed to retrieve a list of active users (<https://api.openpath.com/orgs/9999/users>). The request will always receive the entire population of active OpenPath users. CP will examine the entry returned and extract the name fields and custom field values from each record. CP will then retrieve a list of credentials from the user (<https://api.openpath.com/orgs/9999/users/9999/credentials> ).

If no credentials are found for the user, the user record is ignored.

An example of the data returned for a user is shown below:

{

            "opal": "opal:prod:helium:alpha:5995:user:2502299",

            "id": 2502299,

            "status": "A",

            "startDate": "2021-10-11T06:00:00.000Z",

            "endDate": "2022-01-01T06:59:00.000Z",

            "hasRemoteUnlock": **true**,

            "isOverrideAllowed": **false**,

            "externalId": "AMC PC Services",

            "manuallyInactivatedAt": **null**,

            "lastActivityAt": **null**,

            "createdAt": "2021-08-24T14:31:25.000Z",

            "groups": [

                {

                    "id": 34979,

                    "name": "test 1"

                },

                {

                    "id": 34980,

                    "name": "test 2"

                }

            ],

            "identity": {

                "fullName": "Bernie Adams",

                "initials": "BA",

                "opal": "opal:prod:helium:alpha::identity:2347335",

                "id": 2347335,

                "firstName": "Bernie",

                "middleName": **null**,

                "lastName": "Adams",

                "email": "bernie@amc-hln.com",

                "isEmailVerified": **true**,

                "idpUniqueIdentifier": "",

                "createdAt": "2021-08-24T14:31:25.000Z",

                "updatedAt": "2021-08-27T16:10:34.000Z",

                "namespace": {

                    "id": 7760,

                    "nickname": "org:tiba-parking",

                    "namespaceType": {

                        "id": 2,

                        "name": "Local Org",

                        "modelName": "org"

                    },

                    "org": {

                        "id": 5995,

                        "name": "TIBA Parking"

                    }

                },

                "needsPasswordChange": **false**

            },

            "userCustomFields": [

                {

                    "id": 68261,

                    "value": "302",

                    "customField": {

                        "id": 283,

                        "name": "Company",

                        "customFieldType": {

                            "id": 4,

                            "name": "Dropdown"

                        }

                    }

                },

                {

                    "id": 68277,

                    "value": "304",

                    "customField": {

                        "id": 284,

                        "name": "Sub Company",

                        "customFieldType": {

                            "id": 4,

                            "name": "Dropdown"

                        }

                    }

                },

                {

                    "id": 68253,

                    "value": "298",

                    "customField": {

                        "id": 282,

                        "name": "Parking Profile",

                        "customFieldType": {

                            "id": 4,

                            "name": "Dropdown"

                        }

                    }

                },

                {

                    "id": 97846,

                    "value": **null**,

                    "customField": {

                        "id": 297,

                        "name": "Mobile Number:",

                        "customFieldType": {

                            "id": 1,

                            "name": "Text"

                        }

                    }

                }

            ]

        },

Note the userCustomFields section listing each custom field and, if any, value assigned to the custom field.

The appropriate custom field id and value will be use to look up in the AMC\_SRP\_Database the associated Company ID, Sub Company ID, Access Profile ID or cell phone and associated dropdown item. For any of the custom fields, if the value is blank, null or zero, either ignore the custom field (cell phone, sub-company) or use a default value (Company, Access profile).

For the Company, sub-company and Access profile, it is expected that the left most digits of the description stored in the AMC\_SRP\_Database will have a corresponding accounting ID in the appropriate data field of SmartPark.

For each of the OpenPath users, CP will request the list of credentials from OpenPath (‘users/9999999/credentials’ - list all credentials for specified user Id 9999999). A sample of the data returned is shown below. We will only process the credentials with a credentialType.id = 2 for Wiegand cards.  
{

    "data": [

        {

            "id": 1889534,

            "startDate": "2021-08-01T06:00:00.000Z",

            "endDate": "2050-12-31T07:00:00.000Z",

            "createdAt": "2021-08-26T18:34:20.000Z",

            "updatedAt": "2021-08-26T18:34:20.000Z",

            "credentialType": {

                "id": 6,

                "name": "Cloud Key",

                "modelName": "cloudKey"

            },

            "eventAction": **null**,

            "badgeConfig": **null**,

            "cloudKey": {

                "id": 272479,

                "name": "Bernie Adams",

                "maxDurationSeconds": **null**

            }

        },

        {

            "id": 2168447,

            "startDate": "2021-11-01T06:00:00.000Z",

            "endDate": "2022-01-01T07:00:00.000Z",

            "createdAt": "2021-11-09T17:05:06.000Z",

            "updatedAt": "2021-11-09T17:05:06.000Z",

            "credentialType": {

                "id": 2,

                "name": "Card: Wiegand ID",

                "modelName": "card"

            },

            "eventAction": **null**,

            "badgeConfig": **null**,

            "card": {

                "numBits": 26,

                "fields": {

                    "facilityCode": "54",

                    "cardId": "2920"

                },

                "id": 438705,

                "number": "1947160600878514176",

                "isOutputEnabled": **false**,

                "cardFormat": {

                    "id": 5150,

                    "code": "prox26std",

                    "numBits": 26

                }

            }

        },

        {

            "id": 2169651,

            "startDate": **null**,

            "endDate": **null**,

            "createdAt": "2021-11-09T20:22:05.000Z",

            "updatedAt": "2021-11-09T20:22:05.000Z",

            "credentialType": {

                "id": 2,

                "name": "Card: Wiegand ID",

                "modelName": "card"

            },

            "eventAction": **null**,

            "badgeConfig": **null**,

            "card": {

                "numBits": 37,

                "fields": {

                    "facilityCode": "54",

                    "cardId": "2920"

                },

                "id": 438979,

                "number": "1945555822989803520",

                "isOutputEnabled": **false**,

                "cardFormat": {

                    "id": 5128,

                    "code": "prox37transcore",

                    "numBits": 37

                }

            }

        }

    ],

    "meta": {

        "accessToken": {

            "scopeUpdatedAt": "2021-09-14T14:31:47.000Z"

        },

        "siteSpecificAccess": **null**

    },

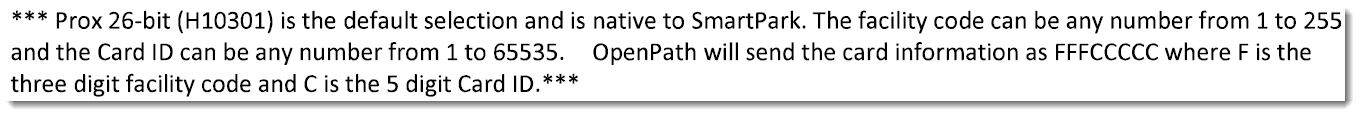
    "totalCount": 3,

    "filteredCount": 3

}

CP will build the badge number from the values found in the card.fields.facilitycode and card.fields.cardId. The resulting badge number will concatenate the facility code and card Id to become the badge number.

From the OpenPath to SmartPark theory document..

 When building the badge number, the card Id will always be padded to 5 digits and then the facility code appended to the front with no padding. For this card:  
"card": {

         "numBits": 26,

         "fields": {

             "facilityCode": "54",

             "cardId": "2920"

        },

The badge number will be: 5402920

If more than one Wiegand credential is returned for a user, all the credentials will be added/updated for the monthly in SmartPark. First credential will be the badge value, second one, badge1 value and so on. Maximum of 5 badges.

CP will utilize the UserId value for the accounting number for monthly lookup in SmartPark.

A log of actions taken will be recorded for each row processed.

At the end of processing, CP will save the value of the last record processed. That saved value will be used for retrieving the next set of user changes.

SmartPark monthly fields affected by CP in adding and updating monthlies from OpenPath:

|  |  |  |
| --- | --- | --- |
| SmartPark Field | OpenPath Field | Notes |
| Company ID | Company – left digits of Desc | Custom Field – blank or 0 - default Used |
| Sub-Company ID | Sub-Company – left digits of Desc | Custom Field - Optional |
| Badge Number | Credential | Facility code + padded card ID |
| Monthly ID | User ID |  |
| First Name | First Name | left 20 chars – SmartPark limit |
| Last Name | Last Name | left 20 chars – SmartPark limit |
| Start Date | Start Date | default today |
| End Date | End Date | default 12/31/2049 |
| Access Profile | Parking Profile – left digits of Desc | Custom Field – blank or 0 – Default used |