

**ACTAtek:  
Access Manager Suite API**

ACTAtek (UK) Ltd.

## **Table of Contents**

<b>Table of Contents</b> .....	<b>3</b>
<b>1 Introduction</b> .....	<b>1</b>
1.1 Purpose .....	1
1.2 Document notations and naming conventions .....	1
1.3 Current AMS API differences .....	1
1.4 AMS API Wsdl File .....	1
<b>2 Objects and Enums</b> .....	<b>2</b>
2.1 SvcResponse .....	2
2.2 SVC_RESPONSE_STATUS.....	2
2.3 Employee .....	3
2.4 FingerPrintImage.....	4
2.5 FP_SECURITY_LEVEL.....	4
2.6 UPDATE_TYPE.....	5
2.7 ADMIN_LEVEL.....	5
2.8 APBTYPE.....	6
2.9 FP_SECURITY_LEVEL.....	6
2.10 EmployeeUpdateCriteria.....	7
2.11 PROPERTY_UPDATE .....	8
2.12 getEmployeeCriteria .....	9
2.13 Department .....	10
2.14 AccessGroup .....	10
2.15 AccessRight.....	11
2.16 TimeScheudle.....	12
2.17 TimeScheudle.....	12
2.18 Event_Log.....	13
2.19 getEvent_LogCriteria.....	14
<b>3 Web Services</b> .....	<b>15</b>
3.1 Login.....	15
3.2 getEmployees .....	16
3.3 AddEmployees .....	16
3.4 UpdateEmployees.....	17
3.5 DeleteEmployees.....	17
3.6 getDepartments .....	18
3.7 AddDepartments .....	18
3.8 UpdateDepartments .....	19
3.9 DeleteDepartments .....	19
3.10 getAccessGroups .....	20
3.11 AddAccessGroups .....	20
3.12 UpdateAccessGroups.....	21
3.13 UpdateAccessGroups.....	21
3.14 getEventLogs.....	22

# 1 Introduction

## 1.1 Purpose

This Document is intended for software developers who have knowledge in programming and web services. The document will cover the objects and web service functions needed to work with the AMS

## 1.2 Document notations and naming conventions

**SC** = Smart card

**PWD** = Password

**FP** = Fingerprint

**AMS** = Access Manager Suite

## 1.3 Current AMS API differences

The AMS API is based around the usage of web services and SOAP as the communication medium between the developer own client application and the AMS software application

The AMS API is a replacement of the previous web service API. The previous web service API was a port over of the old ACTatek API to allow for an easier port over of existing applications.

## 1.4 AMS API WSDL

The **Web Service Definition Language (WSDL)** is an XM based interface definition language that is use for describing the functionality offered by a web service.

The AMS WSDL file is available as part of the AMS installation:

<http://<AMS server IP>/AccessWebAPI/AccessManagerWebAPI.asmx>

This file should be used to generate your client side interface.

## 2 Objects and Enums

This section will provide an overview of the objects used and its properties.

### 2.1 *SvcResponse*

<summary>

This is know as the service response , the [SvcResponse](#) object is used to carry information from the AMS to the client application. It will contain information such as session ID and error/warning messages.

</summary>

<properties>

Message [[string](#)],

- Messages from the AMS

Status [[SVC\\_RESPONSE\\_STATUS](#)]

- Status of the SvcResponse, see [SVC\\_RESPONSE\\_STATUS](#) for more information

SessionID ([string](#)),

- Session ID required for authentication purposes when using the web services

</ properties >

### 2.2 *SVC\_RESPONSE\_STATUS*

<summary>

The [SVC\\_RESPONSE\\_STATUS](#) enumeration used by the [SvcResponse](#) object to indicate the status of the request

</summary>

<properties>

OK [[EnumMember](#)]

- Indicates on issues encountered

ERROR [[EnumMember](#)],

- Indicates an error has occur

LOGINFAIL [[EnumMember](#)]

- Indicates that login to obtain an session has failed

SESSION\_EXPIRED [[EnumMember](#)]

- Indicates that he Session has Exipred

SESSION\_INVALID [[EnumMember](#)]

- Indicates an invaid Session ID has been used

SESSION\_EXPIRED [[EnumMember](#)]

- Indicates that an issue had occurred

</ properties >

## 2.3 Employee

<summary>

The [Employee](#) object contains information of the Users on the AMS

</summary>

<properties>

ID [[int](#)]

- Unique Internal ID of the user

UserID [[string](#)],

- User ID of employee

FirstName [[string](#)],

- Employee first name

LastName [[string](#)],

- Employee Last name

OtherName [[string](#)],

- Employee Other name

Password [[string](#)],

- Password used by employee for login in, empty indicates password is disabled for employee

CardNo [[string](#)],

- Smartcard number assigned to the employee

FpGrpID [[int](#)],

- Fingerprint Group ID

Message [[string](#)],

- Message that appears on ACTatek when employee logs in

UpdateStatus [[UPDATE\\_TYPE](#)]

- Update Status of the [Employee](#), see [UPDATE\\_TYPE](#) for more information

UseCard [[boolean](#)]

- Enable/disable the employee card

Active [[boolean](#)]

- Enable/disable the employee on the ACTatek terminal

UsePassword [[boolean](#)]

- Enable/disable the employee password

UseFingerprint [[boolean](#)]

- Enable/disable the employee fingerprint on the ACTatek terminal

AutoMatch [[boolean](#)]

- Enable/disable the employee fingerprint Automatch on the ACTatek terminal

AutoMatch [[DateTime](#)]

- Sets the employee expiry date on the ACTatek terminal

APB\_Status [[APBTYPE](#)]

- Anti-Passback status of the employee, see [APBTYPE](#) for more information

AdminLvl [[ADMIN\\_LEVEL](#)]

- Admin level of the user on the ACTatek terminal

AdminLvl [[FP\\_SECURITY\\_LEVEL](#)]

- Fingerprint authentication security level on the ACTatek terminal

DepartmentIDList [[List<int>](#)]

- List of Department ID that the employee is assigned to

groupIDList [[List<int>](#)]

- List of Group ID that the employee is assigned to

FLI\_Image [[List<FingerPrintImages>](#)]

- List of fingerprint images, this property is only used if updating the employee fingerprints. It will not contain any fingerprint images if requesting from the AMS

<Employee properties. . . . continue>

```

<Employee properties . . . . continue>
Photo [byte[] ]
- Image of employee
FAM_TEMPLATE [byte[] ]
- FAM fingerprint Template of employee
FLI_TEMPLATE [byte[] ]
- FLI fingerprint Template of employee
hasFLI [boolean]
- Indicates if the employee processes a FLI fingerprint template or not
hasFAM [boolean]
- Indicates if the employee processes a FAM fingerprint template or not
UpdateCriteria [EmployeeUpdateCriteria]
- Update criteria of employee when updating to the AMS, see EmployeeUpdateCriteria for
  more information
</ properties >

```

## 2.4 FingerPrintImage

```

<summary>
The FingerPrintImage object used by the Employee object to store image of a fingerprint which
will then be converted to an FLI template when sent to the AMS
</summary>

```

```

<properties>
FP_IMAGE [byte[]]
- Image of a fingerprint
Remark [EnumMember],
- Remark
</ properties >

```

## 2.5 FP\_SECURITY\_LEVEL

```

<summary>
The FPSECURITY\_LEVEL enumeration used by the Employee object to indicate the fingerprint
authentication security level used on the ACTatek terminal
</summary>
<properties>
LOW [EnumMember]
- Low level
NORMAL [EnumMember],
- Normal level
HIGH [EnumMember]
- High level
HIGHEST [EnumMember]
- Highest level
</ properties >

```

## 2.6 UPDATE\_TYPE

<summary>

The `UPDATE_TYPE` enumeration used by the `Employee` object to indicate the type of update when updating the employee to the AMS

</summary>

<properties>

UPDATE [`EnumMember`]

- Indicates that all employee information needs to be updated

UPDATEPARTIAL [`EnumMember`],

- Indicates that only selected information need to be updated

ADD [`EnumMember`]

- Indicates that is is a new employee

ADDORUPDATE [`EnumMember`]

- Indicates a ADD or UPDATE

</ properties >

## 2.7 ADMIN\_LEVEL

<summary>

The `ADMIN_LEVEL` enumeration used by the `Employee` object to indicate the admin level used on the ACTatek terminal

</summary>

<properties>

PERSONALUSER [`EnumMember`]

- Normal user no admin rights

NETWORKADMIN [`EnumMember`],

- User granted network admin rights

USERADMIN [`EnumMember`]

- User granted user management admin rights

SUPERADMIN [`EnumMember`]

- Super admin , all rights granted

</ properties >

## 2.8 APBTYPE

<summary>

The `APB_TYPE` enumeration used by the `Employee` object to indicate the Ant-Passback type that the employee is under

</summary>

<properties>

DISABLE [[EnumMember](#)]

- Default, Employee is not using APB

AUTOINOUT [[EnumMember](#)],

- Auto In/Out

RES1 [[EnumMember](#)]

- Reserved, not currently in use

REJECT [[EnumMember](#)]

- Reject repeated login

RES2 [[EnumMember](#)]

- Reserved, not currently in use

APB [[EnumMember](#)]

- Anti Passback

LUNCHINOUT [[EnumMember](#)]

- Lunch IN , Lunch OUT

</ properties >

## 2.9 FP\_SECURITY\_LEVEL

<summary>

The `FPSECURITY_LEVEL` enumeration used by the `Employee` object to indicate the fingerprint authentication security level used on the ACTatek terminal

</summary>

<properties>

LOW [[EnumMember](#)]

- Low level

NORMAL [[EnumMember](#)],

- Normal level

HIGH [[EnumMember](#)]

- High level

HIGHEST [[EnumMember](#)]

- Highest level

</ properties >

## 2.10 EmployeeUpdateCriteria

<summary>

The [EmployeeUpdateCriteria](#) object is used when [UPDATE\\_TYPE](#) is set to [UPDATEPARTIAL](#) in the [Employee](#) object, this will allow selective updates of the employee information

</summary>

<properties>

FAM [[boolean](#)]

- Update FAM Template

FLI [[boolean](#)]

- Update FLI Template

FirstName [[boolean](#)]

- Update Employee First Name

LastName [[boolean](#)]

- Update Employee Last Name

Card [[boolean](#)]

- Update Card Number

OtherName [[boolean](#)]

- Update Employee Other Name

ExpiryDate [[boolean](#)]

- Update employee expiry date

FPgrpID [[boolean](#)]

- Update employee Fingerprint Group ID

FPSecurityLvl [[boolean](#)]

- Update employee Fingerprint security level

Message [[boolean](#)]

- Update employee message

Password [[boolean](#)]

- Update employee password

photo [[PROPERTY\\_UPDATE](#)]

- Update employee photo, see [PROPERTY\\_UPDATE](#) for more information

Depts [[PROPERTY\\_UPDATE](#)]

- Update employee department ID list, see [PROPERTY\\_UPDATE](#) for more information

Groups [[PROPERTY\\_UPDATE](#)]

- Update employee group ID list, see [PROPERTY\\_UPDATE](#) for more information

</ properties >

## 2.11 PROPERTY\_UPDATE

<summary>

The `PROPERTY_UPDATE` enumeration used in the `EmployeeUpdateCriteria` for certain properties to indicate the type of update action.

</summary>

<properties>

NONE [EnumMember]

- Default no update action is carried out

ADD\_OR\_UPDATE [EnumMember],

- Add or update new data, if updating a list will merge with existing values

REMOVE\_AND\_UPDATE [EnumMember]

- Remove and then add new data

REMOVE [EnumMember]

- Removes existing data

</ properties >

## 2.12 *getEmployeeCriteria*

<summary>

The `getEmployeeCriteria` object used to specify parameters when querying users from the AMS

</summary>

<properties>

`userIDField` [`string`]

- User ID search value

`partialUserIDField` [`string`]

- Partial User ID search value

`lastNameField` [`string`]

- Last name search value

`firstNameField` [`string`]

- First name search value

`otherNameField` [`string`]

- Other name search value

`departmentIDField` [`string`]

- Department ID search value

`hasDepartmentIDField` [`boolean`]

- Enable/disable the search by department ID

`groupIDField` [`string`]

- Group ID search value

`hasGroupIDField` [`boolean`]

- Enable/disable the search by Group ID

`adminLvlField` [`ADMIN_LEVEL`]

- Admin Level search value

`hasAdminLvlField` [`boolean`]

- Enable/disable the search by user Admin Level

`shiftIDField` [`string`]

- Shift ID search value

`hasShiftField` [`boolean`]

- Enable/disable the search by shift ID

`FPTtypeField` [`TERMINAL_FPTYPE`]

- Fingerprint type search value

`hasFPTtypeField` [`boolean`]

- Enable/disable the search by user Finger print type

`pageSize` [`int`]

- Set size of result set, set to 0 will return all result

`pageIndex` [`string`]

- Set offset of result set, set to 0 will return results from beginning

</ properties >

## 2.13 Department

<summary>

The [Department](#) object used object contains information of the departments in the AMS

</summary>

<properties>

DepartmentID [[int](#)]

- Department ID

DepartmentName [[string](#)],

- Department Name

Description [[string](#)]

- Description of the Department

</ properties >

## 2.14 AccessGroup

<summary>

The [AccessGroup](#) object used object contains information of the Access group in the AMS

</summary>

<properties>

GroupID [[int](#)]

- Access group ID

GroupName [[string](#)],

- Access group Name

Description [[string](#)]

- Description of the Access group

DeptID [[string](#)]

- Department ID that the Access Group belongs to.

ReplaceRights [[boolean](#)]

- Indicate if access rights are to be replaced or merged with existing rights when updating the Access group to the AMS

GroupAccessRight [[AccessRight](#)[]]

- List of Access rights belonging to the Access Group, see [AccessRight](#) for more information

</ properties >

## 2.15 AccessRight

<summary>

The [AccessRight](#) object used object contains information of the access rights in the AMS

</summary>

<properties>

AccessID [[int](#)]

- Access rightID

GroupID [[string](#)],

- Group ID that the Access right belongs to

TerminalID [[string](#)]

- Terminal ID that Access Right belongs to

QuickAccess [[boolean](#)]

- Enable/Disable Quick access on the access right.

AccessRightName [[string](#)]

- Name of the Access right

Mon\_Schedule [[TimeSchedule](#)]

- Access right time schedule for Monday, see [TimeSchedule](#) for more information

Tue\_Schedule [[TimeSchedule](#)]

- Access right time schedule for Tuesday, see [TimeSchedule](#) for more information

Wed\_Schedule [[TimeSchedule](#)]

- Access right time schedule for Wednesday, see [TimeSchedule](#) for more information

Thu\_Schedule [[TimeSchedule](#)]

- Access right time schedule for Thursday, see [TimeSchedule](#) for more information

Fri\_Schedule [[TimeSchedule](#)]

- Access right time schedule for Friday, see [TimeSchedule](#) for more information

Sat\_Schedule [[TimeSchedule](#)]

- Access right time schedule for Saturday, see [TimeSchedule](#) for more information

Sun\_Schedule [[TimeSchedule](#)]

- Access right time schedule for Sunday, see [TimeSchedule](#) for more information

Hol\_Schedule [[TimeSchedule](#)]

- Access right time schedule for Holiday, see [TimeSchedule](#) for more information

</ properties >

## 2.16 TimeScheudle

<summary>

The [TimeSchedule](#) object used object contains the time schedule for the Access Rights  
A single time schedule represents 30 minutes and they will be 48 time schedules per day starting  
at 00:00 and ending at 23:59

</summary>

<properties>

h00\_00\_29 [[Schedule\\_Status](#)]

- Schedule from 00:00 to 00:29, see [Schedule\\_Status](#) for more information

..

..

..

..

h23\_30\_29 [[Schedule\\_Status](#)]

- Schedule from 23:30 to 23:59, see [Schedule\\_Status](#) for more information

</ properties >

## 2.17 TimeScheudle

<summary>

The [Schedule\\_Status](#) enumeration used in the [TimeSchedule](#) to indicate if access is granted or  
denied

</summary>

<properties>

DISABLE [[EnumMember](#)]

- Schedule disabled, access denied

ENABLE [[EnumMember](#)],

- Schedule enable, access granted

</ properties >

## 2.18 Event\_Log

<summary>

The [Event\\_Log](#) object contains information of the event logs in the AMS

</summary>

<properties>

ID [[int](#)]

- Log ID

UserID [[string](#)],

- User ID or user who logged in

FirstName [[string](#)]

- First name of user who logged

LastName [[string](#)]

- Last name of user who logged.

TimestampUTC [[Datetime](#)]

- UTC timestamp of the event log

TimestampLocal [[Datetime](#)]

- Local timestamp, with timezone applied of the event log

Event [[string](#)]

- Event log event, IN, OUT , F1 etc

Timezone [[string](#)]

- Timezone of the event log

TerminalID [[string](#)]

- Terminal ID of where the event log came from

TerminalName [[string](#)]

- Name of terminal of where the event log came from

PhotoCamera[[byte\[\]](#)]

- Image taken when user logged in (for Terminal with camera)

</ properties >

## 2.19 *getEvent\_LogCriteria*

<summary>

The *getEvent\_LogCriteria* object used to specify parameters when querying event logs from the AMS

</summary>

<properties>

ID [[int](#)]

- Log ID

UserID [[string](#)],

- User ID or user who logged in

FirstName [[string](#)]

- First name of user who logged

LastName [[string](#)]

- Last name of user who logged.

TimestampUTC [[Datetime](#)]

- UTC timestamp of the event log

TimestampLocal [[Datetime](#)]

- Local timestamp, with timezone applied of the event log

Event [[string](#)]

- Event log event, IN, OUT , F1 etc

Timezone [[string](#)]

- Timezone of the event log

TerminalID [[string](#)]

- Terminal ID of where the event log came from

TerminalName [[string](#)]

- Name of terminal of where the event log came from

PhotoCamera[[byte\[\]](#)]

- Image taken when user logged in (for Terminal with camera)

</ properties >

## 3 Web Services

### 3.1 Login

<summary>

Login web service is used to establish a login session ID with the AMS , a valid session ID is required when using all other web services

</summary>

<parameters>

pUserID [[string](#)],

- Admin User ID for AMS

pPassword [[string](#)]

- Admin User password for AMS

pClientIP [[string](#)]

- IP address or a unique identifier of PC where client application is installed

</parameters >

<returns>

[SvcReponse](#)

- Service response contain information on actions carried out on the AMS

</returns>

### 3.2 *getEmployees*

<summary>

The web service `getEmployees` is used to query the AMS for its users

</summary>

<parameters>

`pSessionID` [[long](#)],

- Session ID

`pCriteria` [[getEmployeeCriteria](#)]

- User search criteria

</parameters >

<returns>

[Employee](#)[]

- A list of [Employee](#) objects

</returns>

### 3.3 *AddEmployees*

<summary>

The web service `addEmployees` is used to add new employee(s) to the AMS

</summary>

<parameters>

`pSessionID` [[long](#)],

- Session ID

`pEmployeeList` [[Employee](#)[] ]

- List of employees that need to be added to the AMS

</parameters >

<returns>

[SvcResponse](#)

- Service response contain information on actions carried out on the AMS

</returns>

### 3.4 *UpdateEmployees*

<summary>

The web service UpdateEmployees is used to add update existing employee(s) to the AMS

</summary>

<parameters>

pSessionID [[long](#)],

- Session ID

pEmployeeList [[Employee\[\]](#) ]

- List of employees that need to be updated to the AMS

</parameters >

<returns>

[SvcReponse](#)

- Service response contain information on actions carried out on the AMS

</returns>

### 3.5 *DeleteEmployees*

<summary>

The web service DeleteEmployees is used to delete update existing employee(s) to the AMS

</summary>

<parameters>

pSessionID [[long](#)],

- Session ID

pEmployeeList [[string\[\]](#)]

- List of employees user ID that needs to be removed form the AMS

</parameters >

<returns>

[SvcReponse](#)

- Service response contain information on actions carried out on the AMS

</returns>

### 3.6 *getDepartments*

<summary>

The web service `getDepartments` is used to query the AMS for its departments

</summary>

<parameters>

`pSessionID` [[long](#)],

- Session ID

</parameters >

<returns>

[Departments](#) []

- A list of [Department](#) objects

</returns>

### 3.7 *AddDepartments*

<summary>

The web service `AddDepartment` is used to add new departments(s) to the AMS

</summary>

<parameters>

`pSessionID` [[long](#)],

- Session ID

`pDepartmentList` [[Department](#) [] ]

- List of departments that need to be added to the AMS

</parameters >

<returns>

[SvcReponse](#)

- Service response contain information on actions carried out on the AMS

</returns>

### 3.8 *UpdateDepartments*

<summary>

The web service UpdateDepartment is used to update existing departments(s) to the AMS

</summary>

<parameters>

pSessionID [long],

- Session ID

pDepartmentList [Department[] ]

- List of departments that need to be updated to the AMS

</parameters >

<returns>

SvcReponse

- Service response contain information on actions carried out on the AMS

</returns>

### 3.9 *DeleteDepartments*

<summary>

The web service DeleteDepartment is used to delete existing departments(s) to the AMS

</summary>

<parameters>

pSessionID [long],

- Session ID

pIDList [int[]]

- List of departments ID that need to be deleted in the AMS

</parameters >

<returns>

SvcReponse

- Service response contain information on actions carried out on the AMS

</returns>

### 3.10 *getAccessGroups*

<summary>

The web service `getAccessGroups` is used to query the AMS for its Access Groups

</summary>

<parameters>

`pSessionID [long]`,

- Session ID

</parameters >

<returns>

`AccessGroup[]`

- A list of `AccessGroup` objects

</returns>

### 3.11 *AddAccessGroups*

<summary>

The web service `AddAccessGroups` is used to add new group(s) to the AMS

</summary>

<parameters>

`pSessionID [long]`,

- Session ID

`pAccessGroupList [AccessGroup]`,

- Session ID

</parameters >

<returns>

`SvcResponse`

- Service response contain information on actions carried out on the AMS

</returns>

### 3.12 UpdateAccessGroups

<summary>

The web service UpdateAccessGroups is used to update group(s) to the AMS

</summary>

<parameters>

pSessionID [[long](#)],

- Session ID

pAccessGroupList [[AccessGroup](#)],

- Session ID

</parameters >

<returns>

[SvcResponse](#)

- Service response contain information on actions carried out on the AMS

</returns>

### 3.13 UpdateAccessGroups

<summary>

The web service DeleteAccessGroups is used to delete group(s) in the AMS

</summary>

<parameters>

pSessionID [[long](#)],

- Session ID

pIDList [[int](#)],

- Session ID

</parameters >

<returns>

[SvcResponse](#)

- Service response contain information on actions carried out on the AMS

</returns>

### 3.14 *getEventLogs*

<summary>

The web service `getEventLogs` is used to query the AMS for its event logs

</summary>

<parameters>

`pSessionID` [[long](#)],

- Session ID

`pCriteria` [[getEvent\\_LogCriteria](#)],

- Event logs search criteria

</parameters >

<returns>

[Event\\_Log](#)[]

- A list of [Event\\_Log](#) objects

</returns>