

EV[®] Snap

Quick-Start Guide

Account Updater, Controlled Release Version

EVO Account Updater	3
How It Works	3
Integration	4
API Reference	4
Sandbox Plugins	17
Reference Information.....	17

EVO Account Updater

EVO Account Updater works to ensure that the card data on file is the most up to date and accurate, helping reduce merchant risk of losing revenue. This service will assist merchants in updating account information without the time-consuming task of reaching out to customers to manually update account information stored on file. By updating subscription tokens automatically, scaling and throttling account updates, creating efficient infrastructure for easy Wallet account management, and supporting Visa, MasterCard, and Discover card types, EVO Account Updater makes updating and keeping track of card data a Snap*!

How It Works

For merchants using the Snap* Platform Subscription Service and Snap* Tokenization:

If the merchant is using the EVO Snap* platform subscription service, the Snap* Account Updater Internal Service is launched once a day to trigger a CMS query for wallet/subscription tokens for payments that are going to be processed within the next three days. From that information, card data is assembled and then passed through the EVO Account Manager Service to the supported card networks to check that the card data is valid. Finally, EVO sends back an encrypted response file containing the updated card information.

Some merchants use their own applications to handle their subscription or recurring payments processing, and thus can choose to use Snap* tokenization as part of this processing.

For merchants using Snap* Tokenization Only and their own billing engine:

If a merchant decides to use their own software application to manage recurring payments and use the Snap* platform to support tokenization, the merchant application will compile a list of wallet/subscription tokens. The merchant will then pass those tokens to the Snap* platform, which uses AMS to check that the tokens are accurate and up to date. Then, a file with the updated account data information is created, which the merchant application can retrieve.

For merchants storing Card Data and leveraging their own billing engine:

If a merchant chooses to use neither the Snap* platform subscription service nor Snap* tokenization, the merchant can send the information for an account update request as a list of card data, which the Snap* platform can process and return as a list of updated card data.

Integration

1. Merchants will first enroll in EVO Account Manager with their EVO Relationship Manager.
Note: Merchants must be enrolled in each of the three card brands for which they intend to utilize the Account Updater service (Visa, MasterCard, and Discover).
2. Once merchants are enrolled, EVO boarding systems will push their unique Account Updater Merchant IDs to the Snap* platform automatically.
3. Merchants develop their code and will work with EVO Implementation Support as needed to resolve outstanding issues or questions, and once they are ready to test their code and connectivity, they can begin testing via their given certification environment.
Note: No testing is to be done in the Production environment.
4. Once all parties are satisfied that the requirements for certification have been met and successfully tested, a Production environment is set up. Notification is sent to the merchant once the Production environment is ready for use.
5. When Snap* receives an Account Updater request from a merchant, the platform uses the credentials that were assigned and pushed from the enrollment process.
 - * There is a limit of once per day for Account Updater requests.

API Reference

Note: This section only applies to merchants who are using their own recurring billing engine.

REST API Reference

UploadAccountUpdaterRequestFile

RequestUri	https://api.cipcert.goevo.com/2.1.29/REST/DataServices/AMS.svc/Upload?merchantProfileId=<MerchantProfileId> &AUServiceId=<AUServiceId>&fileFormat=<FileFormat>
Method	POST

Note:

- * FileFormat needs to be set based on the format and content of the file being uploaded:
 - o If the file is raw card data in XML format, FileFormat must be set to "AccountData"
 - o If the file is encrypted tokens in XML format, FileFormat must be set to "SnapToken"
 - o If the file is raw card data in JSON format, FileFormat must be set to "AccountDatajson"

- If the file is encrypted tokens in JSON format, FileFormat must be set to "SnapTokenJson"

Request:

Sample XML Request Body for Account Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<AccountDataRequestFile xmlns="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS/AccountManagement">
  <Data>
    <AccountData
      xmlns="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS/AccountManagement">
      <MerchantReferenceId>0000000000</MerchantReferenceId>
      <AccountNumber>5657646586846813</AccountNumber>
      <AccountExpiration>1215</AccountExpiration>
      <CardType>MasterCard</CardType>
      <MerchantProfileId>MerchantId1</MerchantProfileId>
    </AccountData>
    <AccountData
      xmlns="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS/AccountManagement">
      <MerchantReferenceId>0000000001</MerchantReferenceId>
      <AccountNumber>4273253134375833</AccountNumber>
      <AccountExpiration>1215</AccountExpiration>
      <CardType>Visa</CardType>
      <MerchantProfileId>MerchantId1</MerchantProfileId>
    </AccountData>
    <AccountData
      xmlns="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS/AccountManagement">
      <MerchantReferenceId>0000000002</MerchantReferenceId>
      <AccountNumber>6739475637815659</AccountNumber>
      <AccountExpiration>1215</AccountExpiration>
      <CardType>Discover</CardType>
      <MerchantProfileId>MerchantId1</MerchantProfileId>
    </AccountData>
    .
    .
    .
    <AccountData
      xmlns="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS/AccountManagement">
      <MerchantReferenceId>0000000499</MerchantReferenceId>
      <AccountNumber>6725563870129679</AccountNumber>
      <AccountExpiration>1215</AccountExpiration>
      <CardType>Discover</CardType>
      <MerchantProfileId>MerchantId2</MerchantProfileId>
    </AccountData>
  </Data>
  <Summary>
    <TotalItems>1500</TotalItems>
    <MechantSummaries>
      <MerchantSummary>
        <ServiceKey>xyz</ServiceKey>
        <MerchantProfileId>MerchantId1</MerchantProfileId>
        <TotalItems>1000</TotalItems>
      </MerchantSummary>
      <MerchantSummary>
        <ServiceKey>xyz</ServiceKey>
        <MerchantProfileId>MerchantId2</MerchantProfileId>
        <TotalItems>500</TotalItems>
      </MerchantSummary>
    </MechantSummaries>
    <CardTypeSummaries>
      <CardTypeSummary>
```

Sample XML Request Body for Account Data (continued):

```

        <CardType>MasterCard</CardType>
        <TotalItems>883</TotalItems>
    </CardTypeSummary>
    <CardTypeSummary>
        <CardType>Visa</CardType>
        <TotalItems>322</TotalItems>
    </CardTypeSummary>
    <CardTypeSummary>
        <CardType>Discover</CardType>
        <TotalItems>295</TotalItems>
    </CardTypeSummary>
</CardTypeSummaries>
</Summary>
</AccountDataRequestFile>
    
```

Sample JSON Body AccountData File:

```

{
    "Summary" : {
        "MerchantSummaries" : [{
            "MerchantProfileId" : "MerchantId1",
            "TotalItems" : 5
        }],
        "CardTypeSummaries" : [{
            "TotalItems" : 3,
            "CardType" : "MasterCard"
        }, {
            "TotalItems" : 1,
            "CardType" : "Visa"
        }, {
            "TotalItems" : 1,
            "CardType" : "Discover"
        }
        ],
        "TotalItems" : 5
    },
    "Data" : [{
        "AccountExpiration" : "1119",
        "MerchantReferenceId" : "0000000000",
        "AccountNumber" : "5315424430768215",
        "MerchantProfileId" : "MerchantId1",
        "CardType" : "MasterCard"
    }, {
        "AccountExpiration" : "1221",
        "MerchantReferenceId" : "0000000001",
        "AccountNumber" : "4875691479823331",
        "MerchantProfileId" : "MerchantId1",
        "CardType" : "Visa"
    }, {
        "AccountExpiration" : "0420",
        "MerchantReferenceId" : "0000000002",
        "AccountNumber" : "6712654065840296",
        "MerchantProfileId" : "MerchantId1",
        "CardType" : "Discover"
    }
    ],
}
    
```

Sample JSON Body AccountData File (continued):

```

    {
        "AccountExpiration" : "0622",
        "MerchantReferenceId" : 0000000003,
        "AccountNumber" : "2649126955256078",
        "MerchantProfileId" : "MerchantId1",
        "CardType" : "MasterCard"
    }, {
        "AccountExpiration" : "0122",
        "MerchantReferenceId" : 0000000004,
        "AccountNumber" : "2263659360861463",
        "MerchantProfileId" : "MerchantId1",
        "CardType" : "MasterCard"
    }
]

```

Sample XML Request Body for Token Data:

```

<?xml version="1.0" encoding="UTF-8"?>
<TokenDataRequestFile xmlns="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS/AccountManagement">
  <Summary>
    <TotalItems>3</TotalItems>
    <MerchantSummaries>
      <MerchantSummary>
        <ServiceKey>MyServiceKey</ServiceKey>
        <MerchantProfileId>MerchantId1</MerchantProfileId>
        <TotalItems>3</TotalItems>
      </MerchantSummary>
    </MerchantSummaries>
    <CardTypeSummaries>
      <CardTypeSummary>
        <CardType>Visa</CardType>
        <TotalItems>3</TotalItems>
      </CardTypeSummary>
    </CardTypeSummaries>
  </Summary>
  <Data>
    <TokenData>
      <MerchantReferenceId>0000000001</MerchantReferenceId>
      <TokenId>c61b1dcb-9c6b-4700-a0cc-df997cc6d9c2160fcd54-5df2-4969-9342-9fa7f3f2732c</TokenId>
      <CardType>Visa</CardType>
      <MerchantProfileId>MerchantId1</MerchantProfileId>
    </TokenData>
    <TokenData>
      <MerchantReferenceId>0000000002</MerchantReferenceId>
      <TokenId>c61b1dcb-9c6b-4700-a0cc-df997cc6d9c2160fcd54-5df2-4969-9342-9fa7f3f2732c</TokenId>
      <CardType>Visa</CardType>
      <MerchantProfileId>MerchantId1</MerchantProfileId>
    </TokenData>
    <TokenData>
      <MerchantReferenceId>0000000003</MerchantReferenceId>
      <TokenId>c61b1dcb-9c6b-4700-a0cc-df997cc6d9c2160fcd54-5df2-4969-9342-9fa7f3f2732c</TokenId>
      <CardType>Visa</CardType>
      <MerchantProfileId>MerchantId1</MerchantProfileId>
    </TokenData>
  </Data>
</TokenDataRequestFile>

```

Sample JSON Body TokenData File:

```

{
  "Summary" : {
    "TotalItems" : 5,
    "MerchantSummaries" : [{
      "MerchantProfileId" : "MyProfileId1",
      "TotalItems" : 5
    }
  ],
  "CardTypeSummaries" : [{
    "TotalItems" : 2,
    "CardType" : "Visa"
  }, {
    "TotalItems" : 2,
    "CardType" : "MasterCard"
  }, {
    "TotalItems" : 1,
    "CardType" : "Discover"
  }
  ],
  "Data" : [{
    "MerchantProfileId" : "MyProfileId1",
    "CardType" : "Visa",
    "TokenId" : "ecbcf378-8f33-4a60-86b4-75c319b899469b5b89cc-d0ae-461c-b008-edaa636f88e9",
    "MerchantReferenceId" : 000000000
  }, {
    "MerchantProfileId" : "MyProfileId1",
    "CardType" : "Visa",
    "TokenId" : "f7df1650-8345-4f9b-915b-2f4b29e1646a8ce1aafd-ffcf-455b-acfe-7f11c0161317",
    "MerchantReferenceId" : 000000001
  }, {
    "MerchantProfileId" : "MyProfileId1",
    "CardType" : "MasterCard",
    "TokenId" : "573b359c-875c-4a74-b9ec-b6709a32b79898b265ba-4c77-4da6-84f8-2a0a657a286c",
    "MerchantReferenceId" : 000000002
  }, {
    "MerchantProfileId" : "MyProfileId1",
    "CardType" : "MasterCard",
    "TokenId" : "25934448-cf85-4acc-b554-36d0df7397db9f3dd8d3-a6c2-4baf-89d3-f639deeb6b18",
    "MerchantReferenceId" : 000000003
  }, {
    "MerchantProfileId" : "MyProfileId1",
    "CardType" : "Discover",
    "TokenId" : "c5e16206-2e40-4e6d-828e-a2c624e9729dd9ea9795-a5fa-4e6d-bd69-0c382011fa35",
    "MerchantReferenceId" : 000000004
  }
  ]
}

```

Note:

- * The MerchantProfile must be enrolled in Account Updater for one of the three card brands Discover, Visa, and MasterCard (saved on the MerchantProfile)
 - o All merchant profiles used in the file need to be enrolled in Account Updater
- * Merchants need to specify the “accept” header when using JSON to POST messages – if the accept header is not specified, it will default to “application/xml”

- If a merchant is using both XML and an octet-stream, the content type must be set to "content-type: application/octet-stream"
- * MerchantReferenceId is unique by file and is restricted to 10 characters

The table below specifies the different header scenarios and options:

Scenario #1: Upload a file stream using XML, get an XML response
Headers: <ul style="list-style-type: none">• Content-type: "application/octet-stream"• Accept: "application/xml"• Authorization: Basic UEhOaGjXdzZRWE56Wlh..... zTmxjblJwYjI0Kzo=
Scenario #2: Upload a file stream using JSON, get a JSON response
Headers: <ul style="list-style-type: none">• Content-type: "application/octet-stream"• Accept: "application/json"• Authorization: Basic UEhOaGjXdzZRWE56Wlh..... zTmxjblJwYjI0Kzo=

Response:

The response returns two pieces of data:

1. The expected date when the response file will be available for download
2. The RequestId, which is a GUID that the Merchant will use to download the response file

Sample XML Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<UploadResponse xmlns =
"http://schemas.datacontract.org/2004/07/EvoSnap.ServiceFactory.Endpoints.CMS.AccountManagement.Rest"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <ExpectedProcessingDate>2018-11-03T00:00:00+00:00</ExpectedProcessingDate>
  <RequestId>847F2A3273234A79A646CBB5E23EEB40</RequestId>
</UploadResponse>
```

Sample JSON AccountData Response:

```

{
  "Summary" : {
    "TotalItems" : 5,
    "MerchantSummaries" : [{
      "MerchantProfileId" : "MerchantId1",
      "TotalItems" : 4
    }],
    "CardTypeSummaries" : [{
      "CardType" : "MasterCard",
      "TotalItems" : 2
    }, {
      "CardType" : "Visa",
      "TotalItems" : 1
    }, {
      "CardType" : "Discover",
      "TotalItems" : 1
    }
  ]
},
"UpdatedData" : [{
  "MerchantReferenceId" : "0000000002",
  "OldAccountNumber" : "671265XXXXXX0296",
  "NewAccountNumber" : "",
  "OldAccountExpiration" : "0420",
  "NewAccountExpiration" : "",
  "ReasonCode" : "Error",
  "PreviouslySentFlag" : false,
  "ErrorMessage" : "Not enrolled.",
  "CardBrandResponseFile" : "Discover"
}, {
  "MerchantReferenceId" : "0000000001",
  "OldAccountNumber" : "4875691479823331",
  "NewAccountNumber" : "4875691479823331",
  "OldAccountExpiration" : "1221",
  "NewAccountExpiration" : "1221",
  "ReasonCode" : "AccountVerified",
  "PreviouslySentFlag" : true,
  "ErrorMessage" : null,
  "CardBrandResponseFile" : "Visa"
}, {
  "MerchantReferenceId" : "0000000000",
  "OldAccountNumber" : "5315424430768215",
  "NewAccountNumber" : "5213365441254451",
  "OldAccountExpiration" : "1119",
  "NewAccountExpiration" : "1119",
  "ReasonCode" : "AccountNumberUpdate",
  "PreviouslySentFlag" : false,
  "ErrorMessage" : null,
  "CardBrandResponseFile" : "MasterCard"
}, {
  "MerchantReferenceId" : "0000000003",
  "OldAccountNumber" : "2649126955256078",
  "NewAccountNumber" : "2649126955256078",
  "OldAccountExpiration" : "0622",
  "NewAccountExpiration" : "0624",
  "ReasonCode" : "ExpirationDateUpdate",
  "PreviouslySentFlag" : false,
  "ErrorMessage" : null,
  "CardBrandResponseFile" : "MasterCard"
}
]
}

```

Sample JSON TokenData Response:

```
{
  "Summary" : {
    "TotalItems" : 5,
    "MerchantSummaries" : [{
      "MerchantProfileId" : "MerchantId1",
      "TotalItems" : 5
    }],
    "CardTypeSummaries" : [{
      "CardType" : "Visa",
      "TotalItems" : 2
    }, {
      "CardType" : "MasterCard",
      "TotalItems" : 2
    }, {
      "CardType" : "Discover",
      "TotalItems" : 1
    }
  ]
},
"UpdatedData" : [{
  "MerchantReferenceId" : "0000000004",
  "OldMaskedAccountNumber" : "661734XXXXXX6039",
  "NewMaskedAccountNumber" : "",
  "OldAccountExpiration" : "0722",
  "NewAccountExpiration" : "",
  "ReasonCode" : "Error",
  "PreviouslySentFlag" : false,
  "ErrorMessage" : "Not enrolled.",
  "TokenId" : null,
  "CardBrandResponseFile" : "Discover"
}, {
  "MerchantReferenceId" : "0000000000",
  "OldMaskedAccountNumber" : "434032XXXXXX0702",
  "NewMaskedAccountNumber" : "434032XXXXXX0702",
  "OldAccountExpiration" : "0421",
  "NewAccountExpiration" : "0421",
  "ReasonCode" : "AccountVerified",
  "PreviouslySentFlag" : true,
  "ErrorMessage" : null,
  "TokenId" : "ecbcf378-8f33-4a60-86b4-75c319b899469b5b89cc-d0ae-461c-b008-edaa636f88e9",
  "CardBrandResponseFile" : "Visa"
}, {
  "MerchantReferenceId" : "0000000001",
  "OldMaskedAccountNumber" : "494034XXXXXX9162",
  "NewMaskedAccountNumber" : "482614XXXXXX5678",
  "OldAccountExpiration" : "0818",
  "NewAccountExpiration" : "0818",
  "ReasonCode" : "AccountNumberUpdate",
  "PreviouslySentFlag" : true,
  "ErrorMessage" : null,
  "TokenId" : "f7df1650-8345-4f9b-915b-2f4b29e1646a8ce1aafd-ffcf-455b-acfe-7f11c0161317",
  "CardBrandResponseFile" : "Visa"
}, {
  "MerchantReferenceId" : "0000000002",
  "OldMaskedAccountNumber" : "592735XXXXXX2010",
  "NewMaskedAccountNumber" : "592735XXXXXX2010",
  "OldAccountExpiration" : "0818",
  "NewAccountExpiration" : "0521",
  "ReasonCode" : "ExpirationDateUpdate",
  "PreviouslySentFlag" : false,
  "ErrorMessage" : null,
  "TokenId" : "573b359c-875c-4a74-b9ec-b6709a32b79898b265ba-4c77-4da6-84f8-2a0a657a286c",
  "CardBrandResponseFile" : "MasterCard"
}
}
```

Sample JSON TokenData Response (continued):

```

        {
            "MerchantReferenceId" : "0000000003",
            "OldMaskedAccountNumber" : "556952XXXXXX5068",
            "NewMaskedAccountNumber" : "556952XXXXXX5068",
            "OldAccountExpiration" : "0222",
            "NewAccountExpiration" : "0222",
            "ReasonCode" : "AccountVerified",
            "PreviouslySentFlag" : false,
            "ErrorMessage" : null,
            "TokenId" : "25934448-cf85-4acc-b554-36d0df7397db9f3dd8d3-a6c2-4baf-89d3-f639deeb6b18",
            "CardBrandResponseFile" : "MasterCard"
        }
    ]
}
    
```

Note:

- * If the FileFormat on the Request is either "AccountData" or "SnapToken", the response returned will be XML.
- * If the FileFormat on the Request is either "AccountDatajson" or "SnapTokenjson", the response returned will be JSON.

DownloadAccountUpdaterResponseFile

RequestUri	https://api.cipcert.goevo.com/2.1.29/REST/DataServices/AMS.svc/Download/<MerchantProfileId>/<RequestId>
Method	GET

Request:

- * The MerchantProfile must be enrolled in Account Updater for at least one of the three card brands Discover, Visa, and MasterCard (Saved on the MerchantProfile)
 - o All merchant profiles used in the file need to be enrolled in Account Updater
- * The RequestId is returned on the upload, and it will correspond to the response file when it is available for download – if a merchant tries to download before the response file is available, they will get a message indicating the file is unavailable.
- * The stream format returned on the download will depend on what the FileFormat was on the upload.
 - o If the FileFormat was "AccountData" or "SnapToken", the stream will be XML
 - o If the FileFormat was "AccountDatajson" or "SnapTokenjson", the stream will be JSON
- * The structure of the response stream will be the same structure of the request files.
- * Merchants need to specify the "accept" header when using JSON to GET messages – if the accept header is not specified, it will default to "application/xml".

- If a merchant is using both XML and an octet-stream, the content type must be set to "content-type: application/octet-stream"

The table below specifies the different header scenarios and options:

Scenario #1: Request download using XML, get a file stream
<p>Headers:</p> <ul style="list-style-type: none"> • Content-type: "application/xml" • Accept: "application/octet-stream" • Authorization: Basic UEhOaGjXdzZRWE56Wlh..... zTmxjblJwYjI0Kzo=
Scenario #2: Request download using JSON, get a file stream
<p>Headers:</p> <ul style="list-style-type: none"> • Content-type: "application/json" • Accept: "application/octet-stream" • Authorization: Basic UEhOaGjXdzZRWE56Wlh..... zTmxjblJwYjI0Kzo=

SOAP API Reference

UploadAccountUpdaterRequestFile

Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><SOAP-
ENV:Header><ns1:AUServiceId
xmlns:ns1="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS">ABCDE00001</ns1:AUServiceId><ns1:FileFormat
xmlns:ns1="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS">AccountData</ns1:FileFormat><ns1:MerchantProfileI
d
xmlns:ns1="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS">MerchantId1</ns1:MerchantProfileId><ns1:SessionTo
ken
xmlns:ns1="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS">PHNhbWw6QXN...N1cnRpb24+</ns1:SessionToken></SOAP
-ENV:Header> <SOAP-ENV:Body>
    <AUUploadRequest xmlns="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS">
        <Contents>PD94bWwgdmV...0RmlsZT4=</Contents>
    </AUUploadRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Note:

- * SessionToken, MerchantProfileId, and FileFormat are set on the SOAP header.
- * The file is a base 64 encoded onto the body of the request in "Contents".

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  >
  <s:Header>
    <h:ExpectedProcessingDate
      xmlns:h="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS">2018-05-
18T00:00:00+00:00</h:ExpectedProcessingDate>
    <h:RequestId
      xmlns:h="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS">30C5A25D119344DA8C6DC3ACFC0E750D</h:RequestId>
    <ActivityId
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">00000000-0000-0000-0000-
000000000000</ActivityId>
  </s:Header>
  <s:Body>
    <UploadResponse xmlns="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS"/>
  </s:Body>
</s:Envelope>
```

Note:

- * The returned base-64 encoded file can be very large, thus merchants should ensure their settings are set up appropriately to handle such a large value.

DownloadAccountUpdaterRequestFile

Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header>
    <ns1:MerchantProfileId
xmlns:ns1="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS">MerchantId1</ns1:MerchantProfile
Id>
    <ns1:RequestId
xmlns:ns1="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS">30C5A25D119344DA8C6DC3ACFC0E750D
</ns1:RequestId>
    <ns1:SessionToken
xmlns:ns1="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS">PHNhbWw6...cnRpb24+</ns1:Session
Token>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <AUPDownloadRequest
xmlns="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ExpectedProcessingDate
xmlns:h="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS">0001-01-
01T00:00:00</h:ExpectedProcessingDate>
  </s:Header>
  <s:Body>
    <DownloadResponse xmlns="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS">
      <Contents>PD944bWwgdm...RmlsZT4=</Contents>
    </DownloadResponse>
  </s:Body>
</s:Envelope>
```

Base64 Decoded Contents from Download:

```
<?xml version="1.0" encoding="utf-16"?>
<AccountDataResponseFile xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.evosnap.com/CWS/v2.0/DataServices/CMS/AccountManagement">
  <Summary>
    <TotalItems>5</TotalItems>
    <MerchantSummaries>
      <MerchantSummary>
        <MerchantProfileId>MerchantId1</MerchantProfileId>
        <TotalItems>4</TotalItems>
      </MerchantSummary>
    </MerchantSummaries>
```

Base64 Decoded Contents from Download (continued):

```

<CardTypeSummaries>
  <CardTypeSummary>
    <CardType>MasterCard</CardType>
    <TotalItems>2</TotalItems>
  </CardTypeSummary>
  <CardTypeSummary>
    <CardType>Visa</CardType>
    <TotalItems>1</TotalItems>
  </CardTypeSummary>
  <CardTypeSummary>
    <CardType>Discover</CardType>
    <TotalItems>1</TotalItems>
  </CardTypeSummary>
</CardTypeSummaries>
</Summary>
<UpdatedData>
  <UpdatedAccountData>
    <MerchantReferenceId>0000000002</MerchantReferenceId>
    <OldAccountNumber>4274886428819819</OldAccountNumber>
    <NewAccountNumber>4274886428819819</NewAccountNumber>
    <OldAccountExpiration>1215</OldAccountExpiration>
    <NewAccountExpiration>1215</NewAccountExpiration>
    <ReasonCode>AccountVerified</ReasonCode>
    <PreviouslySentFlag>true</PreviouslySentFlag>
    <CardBrandResponseFile>Visa</CardBrandResponseFile>
  </UpdatedAccountData>
  <UpdatedAccountData>
    <MerchantReferenceId>0000000001</MerchantReferenceId>
    <OldAccountNumber>2080101322490441</OldAccountNumber>
    <NewAccountNumber>2080101322490441</NewAccountNumber>
    <OldAccountExpiration>1215</OldAccountExpiration>
    <NewAccountExpiration>1215</NewAccountExpiration>
    <ReasonCode>AccountVerified</ReasonCode>
    <PreviouslySentFlag>false</PreviouslySentFlag>
    <CardBrandResponseFile>MasterCard</CardBrandResponseFile>
  </UpdatedAccountData>
  <UpdatedAccountData>
    <MerchantReferenceId>0000000000</MerchantReferenceId>
    <OldAccountNumber>3626723647197736</OldAccountNumber>
    <NewAccountNumber>3626723647197736</NewAccountNumber>
    <OldAccountExpiration>1215</OldAccountExpiration>
    <NewAccountExpiration>1215</NewAccountExpiration>
    <ReasonCode>AccountVerified</ReasonCode>
    <PreviouslySentFlag>false</PreviouslySentFlag>
    <CardBrandResponseFile>MasterCard</CardBrandResponseFile>
  </UpdatedAccountData>
  <UpdatedAccountData>
    <MerchantReferenceId>0000000003</MerchantReferenceId>
    <OldAccountNumber>2665164434400619</OldAccountNumber>
    <NewAccountNumber>2665164434400619</NewAccountNumber>
    <OldAccountExpiration>1215</OldAccountExpiration>
    <NewAccountExpiration>1215</NewAccountExpiration>
    <ReasonCode>AccountVerified</ReasonCode>
    <PreviouslySentFlag>false</PreviouslySentFlag>
    <CardBrandResponseFile>Discover</CardBrandResponseFile>
  </UpdatedAccountData>
</UpdatedData>
</AccountDataResponseFile>

```

Note:

- * The returned base-64 encoded file can be very large, thus merchants should ensure their settings are set up appropriately to handle such a large value.

Sandbox Plugins

Account Updater Sandbox plugins allow the merchants to test account data going back into XDT. Certain trigger values exist that allow the user to return specific card responses.

The following table defines the current Sandbox plugin trigger values:

Condition Returned	VISA PAN	MasterCard PAN	Discover PAN
Account Number Update	4111-1111-1111-1111	5111-1111-1111-1111	6111-1111-1111-1111
Account Closed	4222-2222-2222-2222	5222-2222-2222-2222	6222-2222-2222-2222
Expiration Date Update	4333-3333-3333-3333	5333-3333-3333-3333	5333-3333-3333-3333
Account Verified	4444-4444-4444-4444	5444-4444-4444-4444	6444-4444-4444-4444
Cardholder Withheld Updates	4555-5555-5555-5555	5555-5555-5555-5555	6555-5555-5555-5555

Reference Information

For additional information, please visit the EVO Snap* Support site at <http://www.evosnap.com/support/> or contact your EVO Technical Support representative.