eContract API for Dealers

Version 1.4

Table of Contents

eContract API for Dealers

Description

The eContract API is used to allow the creation of contracts for dealers and Monitronics accessible from a variety of platforms and methods.

Sources

Name	Description
eContract API	The eContract API Web Service is to be used by anyone that needs to create a contract for monitoring and installation using the contracts supplied by Monitronics.

Security Requirements

The eContractAPI service uses username authentication over transport. The username can be send using soap headers
Here is sample soap header request using WSSE.

```
<soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-1">
        <wsse:Username>user</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">password</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
```

If you do not know your username and password please contact the dealer support team for more information.

eContract API Web Service

Description

The eContract API Web Service is to be used by anyone that needs to create a contract for monitoring and installation using the contracts supplied by Monitronics.

See Also

- Methods
- Complex Types
- Simple Types

Methods: eContract API

Description

These methods are available to all users of the eContract API

Methods

Name	Description
APIVersion	This returns the version of the API that your call is executing against.
AuthenticateUser2	Provides the ability to authenticate a user before they attempt to submit a contract. Also provides meta data for that user which can be used to load certain parameters and discounts.
CreateContract	This method will be used to create embedded or remote signing contracts. When used for embedded signing it will return the embedded signing URL for use within an eContract user interface. If there are two signers both URLs will be returned, but only the first one will really need to be used since when the first signer is done it will automatically launch the second. At the moment the only supported signing permutations are: • Embedded for Primary, None for Secondary • Embedded for Primary and Secondary • Remote for Primary, None for Secondary • Remote for Primary and Secondary
CreateContract2	This method will be used to create embedded or remote signing contracts. It is the same as CreateContract except the ContractData model is changed for this method so that it can include the Discount information. When used for embedded signing it will return the embedded signing URL for use within an eContract user interface. If there are two signers both URLs will be returned, but only the first one will really need to be used since when the first signer is done it will automatically launch the second. At the moment the only supported signing permutations are: • Embedded for Primary, None for Secondary • Embedded for Primary and Secondary • Remote for Primary, None for Secondary • Remote for Primary and Secondary
GetContract	This will return the contract PDF. The PDF will include the Certificate of Completion and the Electronic Disclosure.
GetContractAsPrimary	This will return the contract PDF as the customer would see the contract. The PDF will include the Certificate of Completion and the Electronic Disclosure but not include the sensitive personal information such as credit card numbers.
GetContractID	Retrieves the associated ContractID from the EnvelopeID specified.
SendQuoteEmail	Sends out the quote email for the quote option in the eContract application
SendQuoteEmail2	Sends out the quote email for the quote option in the eContract application using the new ContractData object model with discount support.
VoidContract	This will void an existing contract/envelope

Authentication

In order to use these methods a form of Soap Header insertion is to be implemented.

The eContractAPI service uses username authentication over transport. The username can be send using soap headers
Here is sample soap header request using WSSE.

```
<soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-1">
        <wsse:Username>user</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">password</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
```

If you do not know your username and password please contact the dealer support team for more information.

Method: APIVersion

Description

This returns the version of the API that your call is executing against.

Returns

A APIVersionResult element.

Remarks

This will typically be in a generic date format of when the build had occurred.

Method: AuthenticateUser2

Description

Provides the ability to authenticate a user before they attempt to submit a contract. Also provides meta data for that user which can be used to load certain parameters and discounts.

Parameters

Name	Type	Direction	Description
Login	string	Input	Dealer username
Password	string	Input	Dealer password

Returns

A AuthenticateUser2Result element having the structure defined by the following table.

Name	Type	Description
OptionList	ArrayOfOptionListing	List of options for this user. See the OptionList object for more details.
Result	boolean	true if successful authentication, false otherwise
ResultData	string	message for the result of the call.
dealerDBAs	ArrayOfDealerDBA	Array of DBA's for the dealer this user is associated with.
dealerEmails	ArrayOfDealerEmail	Array of emails for the dealer this user is associated with.
dealerInfo	DealerInfo	Details on the dealer for this user
dealerLicenseExpirationWarning	string	States that the license is nearing expiration
dealerLicenseExpired	string	States where the license has expired
dealerLicenseValid	string	States where the license is valid.
dealerLicenses	ArrayOfDealerLicense	List of licenses for the dealer this user is associated with. First two characters are the state code and the rest is the license number
userInfo	MoniNetUserInfo	Details on the user
Discounts	ArrayOfProgramDiscounts	List of available discounts this user can select from.

Method: CreateContract

Description

This method will be used to create embedded or remote signing contracts. When used for embedded signing it will return the embedded signing URL for use within an eContract user interface. If there are two signers both URLs will be returned, but only the first one will really need to be used since when the first signer is done it will automatically launch the second.

At the moment the only supported signing permutations are:

- Embedded for Primary, None for Secondary
- Embedded for Primary and Secondary
- Remote for Primary, None for Secondary
- Remote for Primary and Secondary

Parameters

Name	Type	Direction	Description
ContractData	ContractDocument	Input	The object model which contains all the information captured for creating the contract.
PrimarySigningType	SigningType	Input	Type of signing to be used for the primary signer. Must be Embedded or Remote. Cannot be None
SecondarySigningType	SigningType	Input	Type of signing to be used for the secondary signer.

Returns

A CreateContractResult element having the structure defined by the following table.

Name	Type	Description
EnvelopeID	string	The envelope ID used to contain all the contract documents.
FaultFields	ArrayOfKeyValueOfstringstring	array of any errors that occurred during the validation process. These are key/value pairs that indicate the field that was being checked and what the issue is with the field.
Result	boolean	true = success, false = failure. when failure, check the ResultData for a message and possibly FaultFields if the error occurred during validation.
ResultData	string	empty when result = true, otherwise the error message.
SigningURL	ArrayOfstring	When embedded signing is used the URL(s) are returned in this list. The first URL is the primary signer URL. The second URL is the secondary signer URL. Typically you will only need the primary signer URL because the system is setup to launch the secondary URL once the primary URL signer is done signing.

Remarks

The return value here is described above.

It is important to note that within the ContractData data type there are two fields (DealerUsername and DealerPassword) which are required to be completed in order to create any kind of contract. The specific validation rules are discussed in the documentation of the ContractData data type.

Method: CreateContract2

Description

This method will be used to create embedded or remote signing contracts. It is the same as CreateContract except the ContractData model is changed for this method so that it can include the Discount information.

When used for embedded signing it will return the embedded signing URL for use within an eContract user interface. If there are two signers both URLs will be returned, but only the first one will really need to be used since when the first signer is done it will automatically launch the second.

At the moment the only supported signing permutations are:

- Embedded for Primary, None for Secondary
- Embedded for Primary and Secondary
- Remote for Primary, None for Secondary
- Remote for Primary and Secondary

Parameters

Name	Type	Direction	Description
ContractData	ContractDocument2	Input	The object model which contains all the information captured for creating the contract.
PrimarySigningType	SigningType	Input	Type of signing to be used for the primary signer. Must be Embedded or Remote. Cannot be None
SecondarySigningType	SigningType	Input	Type of signing to be used for the secondary signer.

Returns

A CreateContract2Result element having the structure defined by the following table.

Name	Type	Description
EnvelopeID	string	The envelope ID used to contain all the contract documents.
FaultFields	ArrayOfKeyValueOfstringstring	array of any errors that occurred during the validation process. These are key/value pairs that indicate the field that was being checked and what the issue is with the field.
Result	boolean	true = success, false = failure. when failure, check the ResultData for a message and possibly FaultFields if the error occurred during validation.
ResultData	string	empty when result = true, otherwise the error message.
SigningURL	ArrayOfstring	When embedded signing is used the URL(s) are returned in this list. The first URL is the primary signer URL. The second URL is the secondary signer URL. Typically you will only need the primary signer URL because the system is setup to launch the secondary URL once the primary URL signer is done signing.

Method: GetContract

Description

This will return the contract PDF. The PDF will include the Certificate of Completion and the Electronic Disclosure.

Parameters

Name	Type	Direction	Description
EnvelopeID	string	Input	The guid for the envelope to retrieve

Returns

A GetContractResult element having the structure defined by the following table.

Name	Type	Description
EnvelopeID	string	The envelope id just downloaded
PDFBytes	base64Binary	The PDF which contains the contract documents.

Method: GetContractAsPrimary

Description

This will return the contract PDF as the customer would see the contract. The PDF will include the Certificate of Completion and the Electronic Disclosure but not include the sensitive personal information such as credit card numbers.

Parameters

Name	Type	Direction	Description
EnvelopeID	string	Input	The guid for the envelope to retrieve

Returns

A GetContractAsPrimaryResult element having the structure defined by the following table.

Name	Type	Description
EnvelopeID	string	The envelope id just downloaded
PDFBytes	base64Binary	The PDF which contains the contract documents.

Method: GetContractID

Description

Retrieves the associated ContractID from the EnvelopeID specified.

Parameters

Name	Type	Direction	Description
EnvelopeID	string	Input	The guid for the envelope to retrieve

Returns

A GetContractIDResult element.

This is just the ContractID of the EnvelopeID that was passed into this method

Method: SendQuoteEmail

Description

Sends out the quote email for the quote option in the eContract application

Parameters

Name	Type	Direction	Description
ContractData	ContractDocument	Input	The completed contract data (payment information is allowed to be missing from this)

Returns

A SendQuoteEmailResult element.

Method: SendQuoteEmail2

Description

Sends out the quote email for the quote option in the eContract application using the new ContractData object model with discount support.

Parameters

Name	Type	Direction	Description
ContractData	ContractDocument2	Input	The completed contract data (payment information is allowed to be missing from this)

Returns

A SendQuoteEmail2Result element.

Remarks

This method saves the quote information and will send out the details of the quote to the customer via email.

Method: VoidContract

Description

This will void an existing contract/envelope

Parameters

Name	Type	Direction	Description
EnvelopeID	string	Input	The guid for the envelope to void

Returns

A VoidContractResult element having the structure defined by the following table.

Name	Type	Description
Result	boolean	true = success, false = failure. when failure, check the ResultData for a message and possibly FaultFields if the error occurred during validation.
ResultData	string	empty when result = true, otherwise the error message.

Complex Types: eContract API

Description

These are the more complex data types that will be passed between the client and server during the use of the eContract Web Service API. The most important one is probably the ContractDocument. This is the core object that will be used to transfer data between client and server.

Complex Types

Name	Description
ArrayOfContactItem	An array of the ContactItem type.
ArrayOfDealerDBA	An Array of possible Dealer DBA (Doing Business As) values. This is primarily needed to ensure the right dealer name is on the contract.
ArrayOfDealerEmail	List of available emails for the dealer.
ArrayOfDealerLicense	List of dealer licenses for a dealer
ArrayOfEquipmentItem	An array of the EquipmentItem type
ArrayOfOptionListing	List of various options that can be used to set default values in the eContact User Interface
ArrayOfProgramDiscounts	List of active discounts available to eContract users
AuthenticationResult2	The object returned by the AuthenticationResult method. Validates the login and returns many useful data elements for the logged in user.
ContactItem	This ContactItem defines the fields needed for each contact assigned to the monitoring account.
ContractDocument	The ContractDocument is the main object to define all aspects of the contract(s) to be created when calling the methods to create new contracts.
ContractDocument2	Extension of the ContractDocument object that supports the new partner discount features of eContract
ContractEnvelope	The data type which stored the envelope ID and the actual PDF bytes of the contract.
CreateContractResult	This is the return object for the CreateContract method.
DealerDBA	The alternate business names for a dealer
DealerEmail	An object used to store a dealer email address and type. Typically this will be contained within an array of these objects.
DealerInfo	List of data elements specific to the dealer that the authenticated user is part of.
DealerLicense	Object used to store the dealer license information
EquipmentItem	The equipment item is used to define the equipment sold and installed for the customer.
MoniNetUserInfo	Information for the specific user that logged into the system with AuthenticateUser2

Name	Description
OptionListing	Various options that can be used to customize the eContract UI. This object is a single option that would normally be part of an array or collection. For example there are several states that do not require a license. An option would be sent down that looks like this: Code = NO_LIC_REQ_STATE CodeDescription = No License Requirement States Country = ALL Description = No License Requirement States Languages = ALL OptionID = 4 Value = CO,ID,IN,KS,KY,MO,ME,NE,NH,OH,SD,VT
PaymentItem	This object defines the necessary fields for capturing payment information. It encapsulates both banking information and credit card information within the same object. Depending on the value of the PaymentType field will determine which fields are required.
ProgramDiscounts	Object used represent an available discount program which can be presented to the end user.
VoidEnvelopeResult	the return object for the VoidEnvelope method

Complex Type: ArrayOfContactItem

Description

An array of the ContactItem type.

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
ContactItem	ContactItem	0..*	This ContactItem defines the fields needed for each contact assigned to the monitoring account. Up to 6 items can be passed in this array.

Remarks

The eContract system supports up to 6 items within this array.

Complex Type: ArrayOfDealerDBA

Description

An Array of possible Dealer DBA (Doing Business As) values. This is primarily needed to ensure the right dealer name is on the contract.

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
DealerDBA	DealerDBA	0..*	Alternate dealer names

Complex Type: ArrayOfDealerEmail

Description

List of available emails for the dealer.

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
DealerEmail	DealerEmail	0..*	Object which contains the email and email type

Remarks

Technically any third party using this API should block access if they do not have an email address that has a Type_ID (see DealerEmail object for more information) of EContract. If more than one email address has a Type_ID of EContract just select the first as the valid email address.

Complex Type: ArrayOfDealerLicense

Description

List of dealer licenses for a dealer

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
DealerLicense	DealerLicense	0..*	DealerLicense object contains information about a specific license the dealer holds. A dealer can have an unlimited number of licenses.

Complex Type: ArrayOfEquipmentItem

Description

An array of the EquipmentItem type

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
EquipmentItem	EquipmentItem	0..*	The equipment item is used to define the equipment sold and installed for the customer. Up to 10 elements can be added to this array.

Remarks

The system supports up to 10 items within this array.

Complex Type: ArrayOfOptionListing

Description

List of various options that can be used to set default values in the eContact User Interface

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
OptionListing	OptionListing	0..*	Array of OptionListing objects

Complex Type: ArrayOfProgramDiscounts

Description

List of active discounts available to eContract users

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
ProgramDiscounts	ProgramDiscounts	0..*	The object that contains all the information for one discount program. There could be two discount types, dollar off or percentage off.

Complex Type: AuthenticationResult2

Description

The object returned by the AuthenticationResult method. Validates the login and returns many useful data elements for the logged in user.

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
OptionList	ArrayOfOptionListing	0..1	List of options for this user. See the OptionList object for more details.
Result	boolean	0..1	true if successful authentication, false otherwise
ResultData	string	0..1	message for the result of the call.
dealerDBAs	ArrayOfDealerDBA	0..1	Array of DBA's for the dealer this user is associated with.
dealerEmails	ArrayOfDealerEmail	0..1	Array of emails for the dealer this user is associated with.
dealerInfo	DealerInfo	0..1	Details on the dealer for this user
dealerLicenseExpirationWarning	string	0..1	States that the license is nearing expiration
dealerLicenseExpired	string	0..1	States where the license has expired
dealerLicenseValid	string	0..1	States where the license is valid.
dealerLicenses	ArrayOfDealerLicense	0..1	List of licenses for the dealer this user is associated with. First two characters are the state code and the rest is the license number
userInfo	MoniNetUserInfo	0..1	Details on the user
Discounts	ArrayOfProgramDiscounts	0..1	List of available discounts this user can select from.

Complex Type: ContactItem

Description

This ContactItem defines the fields needed for each contact assigned to the monitoring account.

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
Ext	string	0..1	Phone number extension if necessary
Name	string	0..1	Full name of the contact to be added.
Password	string	0..1	Password this contact will be using
Phone	string	0..1	Phone number that this contact can be reached at
PhoneType	PhoneTypeEnum	0..1	Type of phone number (ex. Work, Mobile, etc)
UserNumber	string	0..1	Defines the order the contacts appear on the agreement. 1 being the first.

Remarks

The current system supports up to 6 contacts even though the data model technically allows for an unlimited number of contacts.

Complex Type: ContractDocument

Description

The ContractDocument is the main object to define all aspects of the contract(s) to be created when calling the methods to create new contracts.

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
BillStartDate	dateTime	0..1	(REQ) The InstallationStartDate for (non PA/WV states) or InstallationEndDate for all others + the number of Months Paid Up Front + the Promotion Period.
BillingAddress1	string	0..1	(REQ) The first line of the street address to be used for billing purposes.
BillingAddress2	string	0..1	The second line of the street address to be used for billing purposes.
BillingCity	string	0..1	(REQ) The city to be used for billing purposes
BillingCounty	string	0..1	The county to be used for billing purposes. This is required for US contracts, but not Canadian contracts.
BillingState	StateProvinceEnum	0..1	(REQ) The state to be used for billing purposes
BillingZip	string	0..1	(REQ) The zip code or postal code to be used for billing purposes.
CompanyName	string	0..1	(REQ, when the CustomerType = Commerical) Name of the company to use on the contract.
CompanyType	CompanyTypes	0..1	(REQ when the CustomerType = Commerical) Needs to be set to one of the enumerated values of CompanyTypes
ContactList	ArrayOfContactItem	0..1	An array of the ContactItem type
ContractID	int	0..1	(READONLY) This will be supplied upon the creation of a contract
CountryOfSale	CountryEnum	0..1	(REQ) The country in which the contract is being sold.
CustomerType	CustomerTypeEnum	0..1	(REQ) The customer types will be used to determine some of the business rules based on Residential versus Commercial/Business
DealerPassword	string	0..1	(REQ) The password of the dealer or sales rep creating the contract
DealerPersonID	string	0..1	This is the State ID for the dealer that is logged in and using the eContract system. It is not required.

Component	Type	Occurs	Description
DealerRedirectionURL	string	0..1	This option allows for specification as to where the embedded signing returns. It be specified as an URL like http://monitronics.com/. Additional parameters will be added to the end of the URL in order to assist in showing the right messages as returned from the signing room. The events that are possible are: OnSigningComplete = "?eventname=signcomplete" OnViewingComplete = "?eventname=viewcomplete" OnCancel = "?eventname=cancel" OnDecline = "?eventname=decline" OnSessionTimeout = "?eventname=timeout" OnTTLExpired = "?eventname=ttlexpired" OnIdCheckFailed = "?eventname=idcheck" OnAccessCodeFailed = "?eventname=accesscode"; OnException = "?eventname=exception" These are DocuSign events. Some additional information can be found about these events at this link: http://bit.ly/DSEventCodes
DealerUsername	string	0..1	(REQ) The username of the dealer (or sales rep) that is creating the contract.
DraftDay	int	0..1	(REQ) Day of the month that the monthly payment will be made. Valid values for this field are 1 through 28.
EnvelopeID	string	0..1	(READONLY) Returns the DocuSign envelopeID just created.
EquipmentAlarmNetwork	AlarmNetworkEnum	0..1	(REQ, when EquipmentAlarmNetworkIncluded = true) The alarm network to be created with this account. It is possible to specify None here.
EquipmentAlarmNetworkIncluded	boolean	0..1	(REQ) Whether an alarm network is needed.
EquipmentList	ArrayOfEquipmentItem	0..1	List of equipment to be installed and monitored.
EquipmentOtherAmount	decimal	0..1	Any additional amount to be charged to the customer would be supplied here.
EquipmentPermitAmount	decimal	0..1	Any additional amount for permits needed during the installation.

Component	Type	Occurs	Description
EquipmentSubtotalAmount	decimal	0..1	Subtotal of the equipment cost (this will likely be a calculated field)
EquipmentTaxAmount	decimal	0..1	The amount of tax to charge for the equipment for the installation.
EquipmentTotalAmount	decimal	0..1	The total amount to charge for the equipment for the installation (includes all other subtotals and totals)
GuardAddendumRequired	boolean	0..1	When this is true it will include the GuardAddendum document in the contract envelope
InstallationDate	dateTime	0..1	(not currently used)
InstallationFinish	dateTime	0..1	(REQ for PA and CA) Finish date of the installation
InstallationStart	dateTime	0..1	(REQ) Start date for the installation for PA and CA. For all other states it defines when the billing can be started (effectively the installation date).
InstallationWorkDescription	string	0..1	(REQ for PA and CA) Description of the work to be performed during installation.
InsurancePersonalInjuryAmount	decimal	0..1	(REQ for PA) The amount of the personal injury coverage being supplied
InsurancePropertyDamageAmount	decimal	0..1	(REQ for PA) The amount of the property damage coverage being supplied
Language	ContractLanguageEnum	0..1	(REQ) Contract language.
MonthsPaidUpFront	int	0..1	(REQ) The number of months that are being paid up front on the monitoring agreement. For ADP dealers the value can be from 0 to 6. For MOD dealers it has to be zero.
PaymentCount	int	0..1	(REQ, see rules at end of description) The number of payments for the contract (typically 36 or 60). When state = NY or WV this field must be 36. For all other states it must be 36 or 60.
PaymentEffectiveDate	dateTime	0..1	(READONLY) When the payments should start. This is set to be the date of the contract creation.
PaymentExtendedServiceOption	boolean	0..1	(REQ) If the customer wants the Extended Service Option
PaymentInitial	PaymentItem	0..1	(REQ, see rules on PaymentItem type) The payment information for the initial payment

Component	Type	Occurs	Description
PaymentMonthly	PaymentItem	0..1	(REQ, see rules on PaymentItem type) The payment information for the monthly payment
PaymentMonthlyMonitoringRate	decimal	0..1	(REQ) The amount to charge for the monthly monitoring.
PaymentOneTimeActivationFee	decimal	0..1	(REQ) The amount to charge for activation. If no charge, must be zero.
PremiseAddress1	string	0..1	(REQ) Street address of the premise to be monitored and installed.
PremiseAddress2	string	0..1	Second line of street address of the premise to be monitored and installed.
PremiseCity	string	0..1	(REQ) City of the premise to be monitored and installed.
PremiseCounty	string	0..1	(REQ if the country = US) County of the premise to be monitored and installed.
PremiseGateCode	string	0..1	Used for the Guard Addendum (typically this is required for US contracts)
PremiseState	StateProvinceEnum	0..1	(REQ) State of the premise to be monitored and installed.
PremiseZip	string	0..1	(REQ) Zip of the premise to be monitored and installed.
PrimaryBirthDate	string	0..1	Birthdate of the primary signer
PrimaryEmail	string	0..1	(REQ) Email address of the primary signer
PrimaryFirstName	string	0..1	(REQ) First name of the primary signer
PrimaryLastName	string	0..1	(REQ) Last name of the primary signer
PrimaryPassword	string	0..1	(REQ) Password to be used for the alarm system
PrimaryPhone	string	0..1	(REQ) Phone for the primary signer
PrimaryTaxIDNumber	string	0..1	Tax ID / SSN for the primary signer
PromotionPeriod	int	0..1	(REQ) The number of months which are being offered as a promotional period. Valid values are 0, 1, 2, 3. This number combined with MonthsPaidUpFront cannot be greater than 6.
SecondaryBirthDate	string	0..1	Birthdate of the secondary signer
SecondaryEmail	string	0..1	(REQ if there is a secondary signer) Email address of the secondary signer

Component	Type	Occurs	Description
SecondaryFirstName	string	0..1	(REQ if there is a secondary signer)First name of the secondary signer
SecondaryLastName	string	0..1	(REQ if there is a secondary signer)The last name of the secondary signer
SecondaryPhone	string	0..1	The phone number for the secondary signer
SecondaryTaxIDNumber	string	0..1	The Tax ID/SSN for the secondary signer
SurveyCancellingService	boolean	0..1	(REQ) Answer to the question about cancelling existing service during the creation of contract
SurveyConfirmContractLength	boolean	0..1	(REQ) Answer to the question confirming knowledge of the length of the contract
SurveyFamiliarizationPeriod	boolean	0..1	(REQ but only in Canada) Answer to the question about the desire to have a familiarization period
SurveyHomeowner	boolean	0..1	(REQ) Answer to question about the primary signer being the homeowner
SurveyNewConstruction	boolean	0..1	(REQ) Answer to question about the installation being in a newly constructed home
SurveyUnderContract	boolean	0..1	(REQ) Answer to the question about being under and existing contract.

Referenced By

Name	Type
ContractDocument2	Complex Type

Complex Type: ContractDocument2

Description

Extension of the ContractDocument object that supports the new partner discount features of eContract

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
BillStartDate	dateTime	0..1	(REQ) The InstallationStartDate for (non PA/WV states) or InstallationEndDate for all others + the number of Months Paid Up Front + the Promotion Period.
BillingAddress1	string	0..1	(REQ) The first line of the street address to be used for billing purposes.
BillingAddress2	string	0..1	The second line of the street address to be used for billing purposes.
BillingCity	string	0..1	(REQ) The city to be used for billing purposes
BillingCounty	string	0..1	The county to be used for billing purposes. This is required for US contracts, but not Canadian contracts.
BillingState	StateProvinceEnum	0..1	(REQ) The state to be used for billing purposes
BillingZip	string	0..1	(REQ) The zip code or postal code to be used for billing purposes.
CompanyName	string	0..1	(REQ, when the CustomerType = Commerical) Name of the company to use on the contract.
CompanyType	CompanyTypes	0..1	(REQ when the CustomerType = Commerical) Needs to be set to one of the enumerated values of CompanyTypes
ContactList	ArrayOfContactItem	0..1	An array of the ContactItem type
ContractID	int	0..1	(READONLY) This will be supplied upon the creation of a contract
CountryOfSale	CountryEnum	0..1	(REQ) The country in which the contract is being sold.
CustomerType	CustomerTypeEnum	0..1	(REQ) The customer types will be used to determine some of the business rules based on Residential versus Commercial/Business
DealerPassword	string	0..1	(REQ) The password of the dealer or sales rep creating the contract
DealerPersonID	string	0..1	This is the State ID for the dealer that is logged in and using the eContract system. It is not required.

Component	Type	Occurs	Description
DealerRedirectionURL	string	0..1	This option allows for specification as to where the embedded signing returns. It be specified as an URL like http://monitronics.com/. Additional parameters will be added to the end of the URL in order to assist in showing the right messages as returned from the signing room. The events that are possible are: OnSigningComplete = "?eventname=signcomplete" OnViewingComplete = "?eventname=viewcomplete" OnCancel = "?eventname=cancel" OnDecline = "?eventname=decline" OnSessionTimeout = "?eventname=timeout" OnTTLExpired = "?eventname=ttlexpired" OnIdCheckFailed = "?eventname=idcheck" OnAccessCodeFailed = "?eventname=accesscode"; OnException = "?eventname=exception" These are DocuSign events. Some additional information can be found about these events at this link: http://bit.ly/DSEventCodes
DealerUsername	string	0..1	(REQ) The username of the dealer (or sales rep) that is creating the contract.
DraftDay	int	0..1	(REQ) Day of the month that the monthly payment will be made. Valid values for this field are 1 through 28.
EnvelopeID	string	0..1	(READONLY) Returns the DocuSign envelopeID just created.
EquipmentAlarmNetwork	AlarmNetworkEnum	0..1	(REQ, when EquipmentAlarmNetworkIncluded = true) The alarm network to be created with this account. It is possible to specify None here.
EquipmentAlarmNetworkIncluded	boolean	0..1	(REQ) Whether an alarm network is needed.
EquipmentList	ArrayOfEquipmentItem	0..1	List of equipment to be installed and monitored.
EquipmentOtherAmount	decimal	0..1	Any additional amount to be charged to the customer would be supplied here.
EquipmentPermitAmount	decimal	0..1	Any additional amount for permits needed during the installation.

Component	Type	Occurs	Description
EquipmentSubtotalAmount	decimal	0..1	Subtotal of the equipment cost (this will likely be a calculated field)
EquipmentTaxAmount	decimal	0..1	The amount of tax to charge for the equipment for the installation.
EquipmentTotalAmount	decimal	0..1	The total amount to charge for the equipment for the installation (includes all other subtotals and totals)
GuardAddendumRequired	boolean	0..1	When this is true it will include the GuardAddendum document in the contract envelope
InstallationDate	dateTime	0..1	(not currently used)
InstallationFinish	dateTime	0..1	(REQ for PA and CA) Finish date of the installation
InstallationStart	dateTime	0..1	(REQ) Start date for the installation for PA and CA. For all other states it defines when the billing can be started (effectively the installation date).
InstallationWorkDescription	string	0..1	(REQ for PA and CA) Description of the work to be performed during installation.
InsurancePersonalInjuryAmount	decimal	0..1	(REQ for PA) The amount of the personal injury coverage being supplied
InsurancePropertyDamageAmount	decimal	0..1	(REQ for PA) The amount of the property damage coverage being supplied
Language	ContractLanguageEnum	0..1	(REQ) Contract language.
MonthsPaidUpFront	int	0..1	(REQ) The number of months that are being paid up front on the monitoring agreement. For ADP dealers the value can be from 0 to 6. For MOD dealers it has to be zero.
PaymentCount	int	0..1	(REQ, see rules at end of description) The number of payments for the contract (typically 36 or 60). When state = NY or WV this field must be 36. For all other states it must be 36 or 60.
PaymentEffectiveDate	dateTime	0..1	(READONLY) When the payments should start. This is set to be the date of the contract creation.
PaymentExtendedServiceOption	boolean	0..1	(REQ) If the customer wants the Extended Service Option
PaymentInitial	PaymentItem	0..1	(REQ, see rules on PaymentItem type) The payment information for the initial payment

Component	Type	Occurs	Description
PaymentMonthly	PaymentItem	0..1	(REQ, see rules on PaymentItem type) The payment information for the monthly payment
PaymentMonthlyMonitoringRate	decimal	0..1	(REQ) The amount to charge for the monthly monitoring.
PaymentOneTimeActivationFee	decimal	0..1	(REQ) The amount to charge for activation. If no charge, must be zero.
PremiseAddress1	string	0..1	(REQ) Street address of the premise to be monitored and installed.
PremiseAddress2	string	0..1	Second line of street address of the premise to be monitored and installed.
PremiseCity	string	0..1	(REQ) City of the premise to be monitored and installed.
PremiseCounty	string	0..1	(REQ if the country = US) County of the premise to be monitored and installed.
PremiseGateCode	string	0..1	Used for the Guard Addendum (typically this is required for US contracts)
PremiseState	StateProvinceEnum	0..1	(REQ) State of the premise to be monitored and installed.
PremiseZip	string	0..1	(REQ) Zip of the premise to be monitored and installed.
PrimaryBirthDate	string	0..1	Birthdate of the primary signer
PrimaryEmail	string	0..1	(REQ) Email address of the primary signer
PrimaryFirstName	string	0..1	(REQ) First name of the primary signer
PrimaryLastName	string	0..1	(REQ) Last name of the primary signer
PrimaryPassword	string	0..1	(REQ) Password to be used for the alarm system
PrimaryPhone	string	0..1	(REQ) Phone for the primary signer
PrimaryTaxIDNumber	string	0..1	Tax ID / SSN for the primary signer
PromotionPeriod	int	0..1	(REQ) The number of months which are being offered as a promotional period. Valid values are 0, 1, 2, 3. This number combined with MonthsPaidUpFront cannot be greater than 6.
SecondaryBirthDate	string	0..1	Birthdate of the secondary signer
SecondaryEmail	string	0..1	(REQ if there is a secondary signer) Email address of the secondary signer

Component	Type	Occurs	Description
SecondaryFirstName	string	0..1	(REQ if there is a secondary signer)First name of the secondary signer
SecondaryLastName	string	0..1	(REQ if there is a secondary signer)The last name of the secondary signer
SecondaryPhone	string	0..1	The phone number for the secondary signer
SecondaryTaxIDNumber	string	0..1	The Tax ID/SSN for the secondary signer
SurveyCancellingService	boolean	0..1	(REQ) Answer to the question about cancelling existing service during the creation of contract
SurveyConfirmContractLength	boolean	0..1	(REQ) Answer to the question confirming knowledge of the length of the contract
SurveyFamiliarizationPeriod	boolean	0..1	(REQ but only in Canada) Answer to the question about the desire to have a familiarization period
SurveyHomeowner	boolean	0..1	(REQ) Answer to question about the primary signer being the homeowner
SurveyNewConstruction	boolean	0..1	(REQ) Answer to question about the installation being in a newly constructed home
SurveyUnderContract	boolean	0..1	(REQ) Answer to the question about being under and existing contract.
DiscountAmount	decimal	0..1	The amount deducted from the full RMR in dollars. Even if the discount type is Percent this value will be the dollars deducted from the full RMR amount.
DiscountMemberID	string	0..1	The member ID or promo code for the program (for example if the AARP discount is selected this would be the AARP member number)
DiscountName	string	0..1	Name of the discount as retrieved from the AuthenticateUser2 method.
DiscountProgramID	int	0..1	Discount Program ID of the discount as retrieved from the AuthenticateUser2 method.
FullPriceRMR	decimal	0..1	RMR price before any discounts were taken

Complex Type: ContractEnvelope

Description

The data type which stored the envelope ID and the actual PDF bytes of the contract.

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
EnvelopeID	string	0..1	The envelope id just downloaded
PDFBytes	base64Binary	0..1	The PDF which contains the contract documents.

Complex Type: CreateContractResult

Description

This is the return object for the CreateContract method.

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
EnvelopeID	string	0..1	The envelope ID used to contain all the contract documents.
FaultFields	ArrayOfKeyValueOfstringstring	0..1	array of any errors that occurred during the validation process. These are key/value pairs that indicate the field that was being checked and what the issue is with the field.
Result	boolean	0..1	true = success, false = failure. when failure, check the ResultData for a message and possibly FaultFields if the error occurred during validation.
ResultData	string	0..1	empty when result = true, otherwise the error message.
SigningURL	ArrayOfstring	0..1	When embedded signing is used the URL(s) are returned in this list. The first URL is the primary signer URL. The second URL is the secondary signer URL. Typically you will only need the primary signer URL because the system is setup to launch the secondary URL once the primary URL signer is done signing.

Complex Type: DealerDBA

Description

The alternate business names for a dealer

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
ExtensionData	ExtensionDataObject	0..1	not used
DBAName	string	0..1	The alternate dealer business name

Complex Type: DealerEmail

Description

An object used to store a dealer email address and type. Typically this will be contained within an array of these objects.

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
ExtensionData	ExtensionDataObject	0..1	not used
EmailAddress	string	0..1	email address
Type_ID	string	0..1	type of email address, such as EContract. Technically any third party using this API should block access if they do not have an email address that has a Type_ID of EContract. If more than one email address has a Type_ID of EContract just select the first as the valid email address.

Complex Type: DealerInfo

Description

List of data elements specific to the dealer that the authenticated user is part of.

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
ExtensionData	ExtensionDataObject	0..1	not used
CMDealerNo	string	0..1	alternate dealer number
DealerName	string	0..1	name of dealer
InsuranceCompany	string	0..1	name of insurance company for dealer
InsuranceExpirationDate	dateTime	1..1	expiration date of the insurance policy
InsurancePolicyNo	string	0..1	insurance policy number
InsuranceStatus	string	0..1	current status of the insurance policy
MASDealerNo	int	1..1	internal dealer number
MailingAddress	string	0..1	mailing street address
MailingCity	string	0..1	mailing city
MailingCounty	string	0..1	mailing county
MailingState	string	0..1	mailing state or province
MailingZip	string	0..1	mailing zip code or postal code
MainFax	string	0..1	dealer fax number
MainPhone	string	0..1	dealer phone number
PrimaryMASPin	string	0..1	dealer PIN for MAS
PurchaseDealerNo	string	0..1	alternate dealer number
PurchaseFundingTypeID	string	0..1	not used
ServiceDealerNo	string	0..1	not used
ServiceFax	string	0..1	not used
ServiceFundingTypeID	int	1..1	not used
ServicePhone	string	0..1	not used
ShippingAddress	string	0..1	not used
ShippingCity	string	0..1	not used
ShippingCounty	string	0..1	not used
ShippingState	string	0..1	not used
ShippingZip	string	0..1	not used

Complex Type: DealerLicense

Description

Object used to store the dealer license information

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
ExpireDate	dateTime	0..1	Date the license expires
LicenseNumber	string	0..1	The license number issued to the dealer from the state specified in LicensedState
LicenseType	string	0..1	A value that describes the type of license (not required)
LicensedState	string	0..1	Two letter abbreviation of the State or Province.

Complex Type: EquipmentItem

Description

The equipment item is used to define the equipment sold and installed for the customer.

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
Name	string	0..1	Name of the equipment to be installed
Points	int	0..1	Number of points for that piece of equipment
Price	decimal	0..1	Price of the equipment to be charged to the customer.
Quantity	int	0..1	Number of the items to be installed for the customer.
Total	decimal	0..1	Total cost of the equipment line (this field will typically be calculated automatically and returned within the completed object)

Complex Type: MoniNetUserInfo

Description

Information for the specific user that logged into the system with AuthenticateUser2

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
DealerNumber	int	1..1	dealer number for user
Branches	string	0..1	not used
EmailAddress	string	0..1	email address for user
FirstName	string	0..1	first name for user
LastName	string	0..1	last name for user
Roles	ArrayOfString	0..1	list of roles this user has assigned to them. If the roles does not include eContract the user should be denied from submitting a contract.
IsValidUser	boolean	1..1	not used
ContactNo	int	1..1	not used
APP_ID	int	1..1	application ID. Useful for the common funding API
DealerType	int	1..1	Dealer type code would be 0 or 1

Complex Type: OptionListing

Description

Various options that can be used to customize the eContract UI. This object is a single option that would normally be part of an array or collection.

For example there are several states that do not require a license. An option would be sent down that looks like this:

Code = NO_LIC_REQ_STATE

CodeDescription = No License Requirement States

Country = ALL

Description = No License Requirement States

Languages = ALL

OptionID = 4

Value = CO,ID,IN,KS,KY,MO,ME,NE,NH,OH,SD,VT

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
Code	string	0..1	Code that can be used to make decisions (some examples are TERMS, NO_LIC_REQ_STATE)
CodeDescription	string	0..1	Description of the code
Country	string	0..1	Some options are filtered by country, if so the country code would be here otherwise it will say ALL
Description	string	0..1	Description of what this option is about (for example Terms or No License Requirement States)
FilterIndex	string	0..1	not used
GroupOrState	string	0..1	not used
Language	string	0..1	ALL, ES for spanish, EN for English
OptionID	int	0..1	Unique ID that specifically refers to this option
Value	string	0..1	Value to use in the user interface for this option

Complex Type: PaymentItem

Description

This object defines the necessary fields for capturing payment information. It encapsulates both banking information and credit card information within the same object. Depending on the value of the PaymentType field will determine which fields are required.

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
BankAccountNumber	string	0..1	(REQ if PaymentType = BankAccount) The account number of the checking or savings account to be used for payment.
BankRoutingNumber	string	0..1	(REQ if PaymentType = BankAccount) The ABA rouing number of the bank to be used for payment. This is used in conjuction with BankAccountNumber to process the payment. This field must be at least 9 characters long.
CanadaRoutingBranch	string	0..1	Reserved for future use
CanadaRoutingInstitution	string	0..1	Reserved for future use
CreditCardExpireMonth	int	0..1	(REQ if PaymentType = CreditCard) Credit Card expiration month in numerical form (1 = Jan, 2 = Feb, etc)
CreditCardExpireYear	int	0..1	(REQ if PaymentType = CreditCard)Credit Card expirtion year in 4 digit representation (2016, 2017, etc)
CreditCardNumber	string	0..1	(REQ if PaymentType = CreditCard)15 or 16 digit credit card number depending on the CreditCardType field.
CreditCardType	CreditCardTypeEnum	0..1	(REQ if PaymentType = CreditCard)Credit card type to be used for payment (Visa, Mastercard, etc... see CreditCardTypeEnum for valid values)
PaymentType	PaymentTypeEnum	0..1	(REQ) Payment method to be used. See PaymentTypeEnum for valid options.

Remarks

If PaymentType is CreditCard then all the credit card fields are required. If PaymentType is BankAccount then all the bank fields are required.

Complex Type: ProgramDiscounts

Description

Object used represent an available discount program which can be presented to the end user.

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
DisplayName	string	0..1	Name to be shown in the UI
ProgramCode	string	0..1	Unique name that should never change
ProgramID	int	0..1	Internal unique ID
RMRDiscountAmount	decimal	0..1	Amount the total should reduced by (See RMRIsPercent
RMRDiscountCode	string	0..1	Type of discount, but will likely only ever be RMR
RMRIsPercent	boolean	0..1	When true the discount amount is a percentage otherwise it is a dollar amount
ValidCountries	string	0..1	Comma delimited list of the countries this discount program is valid within
ValidForCommercial	boolean	0..1	If this discount is valid for a commercial customer than this will be true, otherwise false
ValidForResidential	boolean	0..1	If this discount is valid for a residential customer than this will be true, otherwise false

Complex Type: VoidEnvelopeResult

Description

the return object for the VoidEnvelope method

Content Model

Contains elements as defined in the following table.

Component	Type	Occurs	Description
Result	boolean	0..1	true = success, false = failure. when failure, check the ResultData for a message and possibly FaultFields if the error occurred during validation.
ResultData	string	0..1	empty when result = true, otherwise the error message.

Simple Types: eContract API

Description

These are the various types that are used within the API. These are primarily enumerators to define specific values for certain data elements.

Simple Types

Name	Description
AlarmNetworkEnum	The alarm network enumerator is used to define the available alarm "networks" that a customer can link to during the installation. Only one alarm network can be selected per installation.
CompanyTypes	The CompanyTypes enumerator is to be used when the CustomerType = Commercial.
ContractLanguageEnum	Language the contract will be created
CountryEnum	List of valid countries for eContract
CreditCardTypeEnum	The CreditCardTypeEnum will be used to select the type of credit card being used for payment when the PaymentType is CreditCard.
CustomerTypeEnum	The customer types will be used to determine some of the business rules based on Residential versus Commercial/Business.
PaymentTypeEnum	The PaymentTypeEnum determines the type of payment that is being used both for the initial and monthly payments.
PhoneTypeEnum	The PhoneTypeEnum describes the types of phone numbers that can be specified in the ContactList
SigningType	Signing type to implement within the signing room.
StateProvinceEnum	List of all possible states for US and Canada

AlarmNetworkEnum

Description

The alarm network enumerator is used to define the available alarm "networks" that a customer can link to during the installation. Only one alarm network can be selected per installation.

Derived By

Restricting string

Enumeration

Value	Description
AlarmDotcom	Alarm.com value
AlarmNet	Alarmnet value
Tellular	Tellular value
DSC	Digital Security Controls value
icontrol	icontrol value
None	When no alarm network is selected then this value should be selected. It indicates that the user selected something and that something is None.

Simple Type: CompanyTypes

Description

The CompanyTypes enumerator is to be used when the CustomerType = Commercial.

Derived By

Restricting string

Enumeration

Value	Description
Corporation	When the business is any type of corporation
Proprietorship	When the business is a single person
LLC	When the business is a limited liability corporation
Partnership	When the business is a partnership

Simple Type: ContractLanguageEnum

Description

Language the contract will be created

Derived By

Restricting string

Enumeration

Value	Description
English	Will use the English language
Spanish	Will use the Spanish language. NOTE: Spanish contracts are not valid in Canada.

CountryEnum

Description

List of valid countries for eContract

Derived By

Restricting string

Enumeration

Value	Description
US	United States
CA	Canada (NOTE: Only English contracts are valid in Canada)

CreditCardTypeEnum

Description

The CreditCardTypeEnum will be used to select the type of credit card being used for payment when the PaymentType is CreditCard.

Derived By

Restricting string

Enumeration

Value	Description
Visa	Credit card type of Visa (begins with 4)
MasterCard	Credit card type of MasterCard (begins with 5)
Discover	Credit card type of Discover (begins with 6)
AmericanExpress	Credit card type of American Express (begins with 3)

CustomerTypeEnum

Description

The customer types will be used to determine some of the business rules based on Residential versus Commercial/Business.

Derived By

Restricting string

Enumeration

Value	Description
Commercial	If the customer is not a home the contract should be tagged as a Business customer
Residential	If the customer is monitoring a home this value should be used for the CustomerType

PaymentTypeEnum

Description

The PaymentTypeEnum determines the type of payment that is being used both for the initial and monthly payments.

Derived By

Restricting string

Enumeration

Value	Description
Invoice	This would be used for a monthly invoice payment
BankAccount	This would be used for a checking or savings account
CreditCard	This would be used when a credit card will be used for payment.

PhoneTypeEnum

Description

The PhoneTypeEnum describes the types of phone numbers that can be specified in the ContactList

Derived By

Restricting string

Enumeration

Value	Description
Home	Home phone number
Cell	Cell or mobile phone number
Work	Work or business phone number

Simple Type: SigningType

Description

Signing type to implement within the signing room.

Derived By

Restricting string

Enumeration

Value	Description
Embedded	Embedded Signing (Within the browser)
Remote	Remote Signing (via email)
None	No Signing

StateProvinceEnum

Description

List of all possible states for US and Canada

Derived By

Restricting string

Enumeration

Value	Description
AL	Alabama
AK	Alaska
AZ	Arizona
AR	Arkansas
CA	California
CO	Colorado
CT	Connecticut
DE	Delaware
FL	Florida
GA	Georgia
HI	Hawaii
ID	Idaho
IL	Illinois
IN	Indiana
IA	Iowa
KS	Kansas
KY	Kentucky
LA	Louisiana
ME	Maine
MD	Maryland
MA	Massachusetts
MI	Michigan
MN	Minnesota
MS	Mississippi
MO	Missouri
MT	Montana
NE	Nebraska
NV	Nevada
NH	New Hampshire
NJ	New Jersey
NM	New Mexico
NY	New York
NC	North Carolina

Value	Description
ND	North Dakota
OH	Ohio
OK	Oklahoma
OR	Oregon
PA	Pennsylvania
RI	Rhode Island
SC	South Carolina
SD	South Dakota
TN	Tennessee
TX	Texas
UT	Utah
VT	Vermont
VA	Virginia
WA	Washington
WV	West Virginia
WI	Wisconsin
WY	Wyoming
AS	American Samoa
DC	District of Columbia
FM	Federated States of Micronesia
GU	Guam
MH	Marshall Islands
MP	Northern Mariana Islands
PW	Palau
PR	Puerto Rico
VI	Virgin Islands
AB	Alberta
BC	British Columbia
MB	Manitoba
NB	New Brunswick
NL	Newfoundland and Labrador
NS	Nova Scotia
NT	Northwest Territories
NU	Nunavut
ON	Ontario
PE	Prince Edward Island
QC	Quebec
SK	Saskatchewan
YT	Yukon