

Apex EDI Eligibility 2

19 November 2020

1 Introduction

The purpose of eligibility is to verify the medical or dental benefits for which a particular patient is eligible. Requests can be general, or can relate to a specific type of service (surgical, diagnostic X-ray, anesthesia, periodontics, etc.) or procedure code.

1.1 Structure of eligibility request

Eligibility requests are structured as a three- or four-level hierarchy consisting of one or more payers (typically insurance companies) at the first level, each with one or more payees (service providers) at the second level, each with one or more subscribers at the third level. A subscriber may optionally have one or more dependents at the fourth level.

Requested benefits are associated with one or more patients. Each patient appears at either the subscriber (third) level or the dependent (fourth) level of the hierarchy described above, depending on whether the patient is assigned a member ID by the payer (subscriber position) or is identified only as a dependent of an insured person (dependent position).

Note that a patient may technically be a dependent of the insured, but as long as the patient is assigned his or her own member ID, the patient appears in the subscriber position, not the dependent position. A patient only appears in the dependent position in the request hierarchy if the patient does not have his or her own member ID, but instead must be identified by name, birthdate, etc., and as a dependent of a subscriber with a member ID.

1.2 Identification of patients

Note that benefit responses are not necessarily structured the same as the source benefit requests. For example, for a benefit request with multiple patients, there may be multiple benefit responses in which the patient records are grouped together in arbitrary ways.

By far the easiest and most reliable way to identify the benefits associated with a patient is to assign a unique PayeeTraceNumber to each patient for which benefits are requested. Each patient (and the associated benefits) in a response has the same PayeeTraceNumber as the corresponding patient in the request. See below for more details about PayeeTraceNumber.

1.3 Format of eligibility request

The eligibility request contains an array of one or more benefit inquiries. There is also a flag in the request that indicates that the request whether the request is a live or test request.

1.3.1 Eligibility inquiries

An eligibility inquiry is formatted as a collection of nested objects – payers, payees, subscribers, dependents, etc. The eligibility inquiry is closely modeled on the ASC X12 270 Health Care Eligibility Benefit Inquiry and Response standard. Eligibility inquiries are converted to one or more 270 documents for submission to payers.

1.3.2 Multiple eligibility requests vs. multiple eligibility inquiries vs. multiple patients per inquiry

An eligibility request can be as simple as a single inquiry with a single patient. For eligibility requests submitted in real time, there is no need for more complexity. However, if there is a need to submit requests for multiple patients at about the same time, the options include the following:

- Submit multiple requests, each with a single inquiry, where the inquiry contains a single patient.
- Submit a single request with multiple inquiries, where each inquiry contains a single patient.
- Submit a single request with a single inquiry, where the inquiry contains multiple patients.

While all of these options are valid, the following should be kept in mind:

- When requesting the responses for a particular eligibility request, *all* responses for *all* inquiries are returned. If many inquiries and/or patients are included in the request, and one of the responses is delayed for some reason (such as a payer that only responds once a day in batch mode), then it may be necessary to fetch and inspect many successful responses over and over again until the last outstanding response is received. If there is only one patient per request, then once a response for the patient is received, it will not be necessary to ask for and process the response for that patient again.
- The ASC X12 270 Health Care Eligibility Benefit Inquiry and Response standard sets a limit of 99 patients per inquiry. While Apex EDI does not set a predetermined limit to the number of patients in an inquiry, it may be best not to exceed a reasonable limit, such as 99.

Unless there are special circumstances, it will generally be simpler to submit one patient per request.

1.4 Structure of the eligibility response

An eligibility response has the same three- or four-level hierarchy structure as an eligibility request, with the main difference that the patient records contain benefit information (payer responses) instead of benefit requests (from care providers).

1.5 Initial response versus follow-up responses

SubmitEligibilityRequest() returns a response that *may* contain some or all of the responses for the patients included in the request. In order to return as many responses as possible, SubmitEligibilityRequest() will wait before returning until responses are available for all patients in the request or for up to 60 seconds, whichever comes first.

Requests are generally submitted by Apex for immediate response when possible. Even if a request contains multiple patients, the request will be broken up into multiple individual requests if necessary in order to increase the chances of receiving an immediate response.

In spite of this, it is not possible in every case for all responses to be received before SubmitEligibilityRequest() returns. Reasons include:

- Some payers will only respond after a delay, sometimes of several hours. Thus even a request with a single patient may not receive an immediate response.
- If there are multiple patients, it may not be possible for all requests to be submitted and responses returned and processed within the 60 seconds allotted for SubmitEligibilityRequest().

In general, a response will be received for every inquiry, usually in the value returned from `SubmitEligibilityRequest()`, but this is not possible to guarantee, and it is not possible to receive an estimate of when the response will be received. In extreme cases, such as a payer experiencing problems with its eligibility system, a request may be successfully submitted, but no response ever received.

`SubmitEligibilityRequest()` returns an instance of `WsBenefitSubmitResult`, which includes an array of `WsBenefitResponse` objects. If the instance of `WsBenefitResponse` returned by `SubmitEligibilityRequest()` does not contain responses for all patients in the request, then follow-up calls may be made to `GetEligibilityResponses()`. The information returned by `GetEligibilityResponses()` is built up incrementally as responses are received. `GetEligibilityResponses()` will always return all responses returned in any previous call to `SubmitEligibilityRequest()` or `GetEligibilityResponses()`, plus any additional responses that may have been received since the last call to `SubmitEligibilityRequest()` or `GetEligibilityResponses()`. As mentioned above, while all responses will typically be available soon after the initial call to `SubmitEligibilityRequest()`, in some cases responses may be delayed for a few hours or days, and in extreme cases a response for a particular patient may never become available.

1.6 Case sensitivity

Alphabetic characters in responses will generally be upper case, but this is not guaranteed. The requestor should not assume that responses will contain a particular case of characters, or that the case of characters in strings that originate in the request and are returned in the response (such as the `PayeeTraceNumber` described below) will have the case preserved.

2 Return values

The `BenefitSubmitResult` object returned by `SubmitEligibilityRequest()` contains the following information:

- A request ID. This value must be passed in to `GetEligibilityResponses()` to identify the request for which the responses are desired.
- A status. The status indicates whether all inquiries were invalid, or if not, if some responses are still pending, or if all responses have been received.
- A list of responses. A response may be a successful benefit response or may be an error response from the payer.
- A list of validation errors. Validation errors include such issues as missing required information, such as a missing payer ID.
- A list of errors. These include errors such as the inability for of Apex to connect to the payer's eligibility system.

The `BenefitResponseResult` object returned by `GetEligibilityResponses()` is of a different type from `BenefitSubmitResult`, but it contains the same information.

3 Minimal eligibility request

The minimal eligibility request consists of a single inquiry that contains a single payer, a single payee, a single subscriber, and optionally a single dependent.

- `RequestType` must be set to `WsRequestType.Request`.
- The payer must be identified with a valid Apex-assigned `PayerId`. `Type` must be set, typically to `WsPayerType.Payer`.

- The payee must be identified with some sort of ID accepted by the payer, typically NationalProviderId or TaxIdNumber. CommonName (the last name if an individual, or the name of the organization) must be given. If the payee is an individual, then FirstName should be set. EntityType must be set to either WsEntityType.Individual or WsEntityType.Organization. Type must be set, typically to WsPayeeType.Provider, WsPayeeType.Hospital, or WsPayeeType.Facility.
- The subscriber must be identified in some way, usually by the member ID assigned by the payer. Some payers may require additional identification. The name and birthdate of the subscriber must also be given.
- If there is a dependent, then the name and birth date of the dependent must be given. Some payers may require additional identification. The relationship of the dependent to the subscriber may be specified.
- The patient record (whether the subscriber or the dependent) must contain a benefit request identifying either one or more types of service or one or more procedure codes (but not a mixture of the two). The most general request simply includes the most general type of service, WsServiceType.HealthBenefitPlanCoverage.
- It is optional, but very highly recommended, that a trace number be assigned to each patient, as described in the section *Identification of patients*. A trace number may *only* be assigned to a patient (a subscriber or dependent for whom benefits are requested). A trace number must *not* be assigned to a subscriber if no benefits are requested for the subscriber.

4 More complex eligibility requests

While a minimal eligibility request consists of a single inquiry with a single patient and a single service type, more complex requests are supported.

4.1 Multiple patients

There may be multiple inquiries in a single request. An inquiry may contain one or more payers, each of which may contain one or more payees, each of which may contain one or more subscribers, each of which may contain one or more dependents.

If more than one patient (subscriber or dependent) is supplied in the entire request, then it is very highly recommended that trace numbers be assigned to the patients, as described in the section *Identification of patients*. Otherwise, it may be difficult to match the patients in the request from the patients in the response. Note that while each patient is associated with a particular inquiry in the request, the patients are not grouped by inquiry in the response. Trace numbers allow patients to be identified unambiguously.

A subscriber must only be included in an inquiry if benefits are requested for the subscriber (the subscriber is a patient) or if the subscriber has one or more associated dependents with requested benefits; otherwise, the inquiry will be rejected as invalid. Similarly, a dependent must not be included in an inquiry if benefits are requested.

4.2 More complex benefit types

Instead of identifying a single general service type in the benefit request associated with a patient, more complex benefit information may be supplied. The service type may be more specific. Multiple service types (up to 99) may be specified. Alternatively, instead of service types, a procedure code may be specified along with up to four diagnosis codes to support the procedure. Note: Either one or more service types or a procedure code may be specified in a single benefit object, but not both.

Up to 99 benefit objects may be associated with a single patient. Therefore, while a single benefit object will only support one procedure code, information about multiple procedure codes may be requested by associating multiple benefit objects with a single patient, each with a different procedure code.

5 How to match patients in requests to patients in responses

Requests and responses can easily be linked by matching the PayeeTraceNumber of a patient in a request to the PayeeTraceNumber assigned to a patient in a response. Although PayeeTraceNumber is optional, and in some cases it may be possible to link a patient in a request to a patient in a response without assigning a PayeeTraceNumber to the patient, it is highly recommended that PayeeTraceNumber be set for every patient in every request. If multiple patients are submitted in a request, using the PayeeTraceNumber is by far the simplest way of linking the patients in the request with the patients in the response.

PayeeTraceNumber has two required components (Number and OriginatorId) and one optional component (OriginatorSecondaryId). Number can be one to 50 characters in length, but should be unique within the set of request patients to be matched against when responses are received. A good way to generate the Number component would be to use a database sequence to generate unique numbers.

OriginatorId identifies the organization submitting the request. It must be exactly 10 characters long, although it may contain spaces. The first character must be “1” if an EIN is used, “3” if a DUNS is used, or “9” if some other identifier is used. Also, to prevent collisions with trace numbers assigned by Apex EDI, OriginatorId must not be “9APEXEDI99”.

OriginatorSecondaryId is optional. It can be used to identify a department within the requesting organization. If present, it must be one to 50 characters in length.

A trace number may *only* be assigned to a patient (a subscriber or dependent for whom benefits are requested). A trace number must *not* be assigned to a subscriber if no benefits are requested for the subscriber.

6 Test requests

Test requests that are not routed to payers can be made by setting the test flag and specifying one or more test payers in the eligibility request. Only test payers may be specified in a test request, and only live payers may be specified in a live request; otherwise, the entire request will be rejected as invalid.

In order to identify test payers, call the new GetTestPayers() API. Note that at this time, test payers may only be used for eligibility requests, not claim submission.

The member ID of the subscriber controls what will be returned. Note that the member ID of the subscriber governs, even if the benefits are being requested for a dependent.

- 1001 – envelope errors.
- 1002 – payer errors.
- 1003 – payee errors.
- 1004 – subscriber errors.
- 1005 – dependent errors.
- 1006 – syntax error.

- 1007 – complex benefits will be returned.
- 1008 – simple benefits will be returned after a delay of approximately 65 seconds. Note that there will be a 65-second delay for each patient (subscriber or dependent) where 1008 is the subscriber member ID.
- Any other member ID – simple benefits will be returned.

Note that while any member ID other than the ones listed above will return simple benefits, member IDs in the range of 1000 to 1999 are reserved for special return values, so they should be avoided. Member ID 2001 will always be safe to use to successfully generate a benefit response.

7 API details

7.1 SubmitEligibilityRequest

SubmitEligibilityRequest() is used to request eligibility information for one or more patients.

SubmitEligibilityRequest() will block until responses are received for all patients in all valid inquiries, or until approximately 60 seconds have elapsed.

7.1.1 Signature

WsBenefitSubmitResult SubmitEligibilityRequest(string vendorKey, string password, string vendorSiteId, WsBenefitInquiry[] benefitInquiries)

7.1.2 Inputs

vendorKey

Type: String

Description: The unique key assigned to the vendor.

password

Type: String

Description: The vendor password.

vendorSiteId

Type: String

Description: The ID of the vendor site.

benefitInquiries

Type: Array of WsBenefitInquiry.

Description: Each entry contains a request for eligibility results for one or more patients.

7.1.3 Returns

Type: WsBenefitSubmitResult

Description: The object returned contains the request ID assigned by Apex which must be passed in to any subsequent related calls to GetEligibilityResponses(). The object also contains all patient eligibility results and/or error responses that have been received by the time the method returns, validation results, and a status which indicates whether all results are being returned, etc. See below for more details.

7.2 GetEligibilityResponses

If not all responses are received by the time SubmitEligibilityRequest() returns, then subsequent calls may be made to GetEligibilityResponses() to obtain any additional responses that might have been received. Results are cumulative, meaning that any responses returned by any previous call associated with the same request ID to SubmitEligibilityRequest() or GetEligibilityResponses() will continue to be returned in any subsequent call to GetEligibilityResponses(), along with any new responses that might have been received in the mean time.

7.2.1 Signature

WsBenefitResponseResult GetEligibilityResponses(string vendorKey, string password, string requestId, string vendorSiteId)

7.2.2 Inputs

vendorKey

Type: String

Description: The unique key assigned to the vendor.

password

Type: String

Description: The vendor password.

vendorSiteId

Type: String

Description: The ID of the vendor site.

requestId

Type: String

Description: The string identifying the request, returned by Apex from a call to SubmitEligibilityRequest().

7.2.3 Returns

Type: WsBenefitResponseResult

Description: WsBenefitResponseResult has the same members as WsBenefitSubmitResult, with the same meanings. See the documentation for the return value of SubmitEligibilityRequest and the documentation for type type WsBenefitSubmitResult.

7.3 Types

7.3.1 WsBenefitInformationType

public enum WsBenefitInformationType

```
{  
    Unknown,  
    ActiveCoverage,  
    ActiveFullRiskCapitation,  
    ActiveServicesCapitated,  
    ActiveServicesCapitatedToPrimaryCarePhysician,  
    ActivePendingInvestigation,  
    Inactive,  
    InactivePendingEligibilityUpdate,  
    InactivePendingInvestigation,  
    CoInsurance,  
    CoPayment,  
    Deductible,  
    BenefitDescription,  
    Exclusions,  
    Limitations,  
    OutOfPocketStopLoss,  
    Unlimited,  
    NonCovered,  
    CostContainment,  
    Reserve,  
    PrimaryCareProvider,  
}
```

```

    PreExistingCondition,
    ServicesRestrictedToFollowingProvider,
    NotDeemedAMedicalNecessity,
    BenefitDisclaimer,
    SecondSurgicalOpinionRequired,
    OtherOrAdditionalPayor,
    PriorYearSHistory,
    CardSReportedLostStolen,
    ContactFollowingEntityForEligibilityOrBenefitInformation,
    CannotProcess,
    OtherSourceOfData,
    HealthCareFacility,
    SpendDown,
    CoverageBasis,
    ManagedCareCoordinator
}

```

7.3.2 WsBenefitStatus

AllInvalid means that all of the inquiries in the request failed validation, so none will be submitted for further processing. ResultsIncomplete means that there is at least one valid inquiry for which no response has been received yet. ResultsComplete means that responses have been received for all valid inquiries.

```

public enum WsBenefitStatus
{
    Unknown,
    AllInvalid,
    ResultsComplete,
    ResultsIncomplete,}

```

7.3.3 WsCoverageLevel

```

public enum WsCoverageLevel
{
    Unknown,
    ChildrenOnly,
    DependentsOnly,
    EmployeeAndChildren,
    EmployeeOnly,
    EmployeeAndSpouse,
    Family,
    Individual,
    SpouseAndChildren,
    SpouseOnly
}

```

7.3.4 WsDiagnosticCodeType

```

public enum WsDiagnosticCodeType
{
    Unknown,

```



```
    Icd9,  
    Icd10,  
    Snodent  
}
```

7.3.5 [WsEligibilityProcedureCodeSource](#)
public enum WsEligibilityProcedureCodeSource

```
{  
    Unknown,  
    Ada,  
    Cpt,  
    Hcpcs,  
    Icd9,  
    Hiec,  
    Ndc,  
    MutuallyDefined  
}
```

7.3.6 [WsEntityType](#)
public enum WsEntityType

```
{  
    Unknown,  
    Individual,  
    Organization  
}
```

7.3.7 [WsFollowupAction](#)
public enum WsFollowupAction

```
{  
    Unknown,  
    PleaseCorrectAndResubmit,  
    ResubmissionNotAllowed,  
    ResubmissionAllowed,  
    DoNotResubmitInquiryInitiatedToAThirdParty,  
    PleaseWait30DaysAndResubmit,  
    PleaseWait10DaysAndResubmit,  
    DoNotResubmitWeWillHoldYourRequestAndRespondAgainShortly  
}
```

7.3.8 [WsEligibilityProcedureCodeSource](#)
public enum WsEligibilityProcedureCodeSource

```
{  
    Unknown,  
    Ada, // American Dental Association  
    Cpt, // Current Procedural Terminology  
    Hcpcs, // Healthcare Common Procedure Coding System (HCPCS)  
    Icd9, // International Classification of Diseases, 9th Revision, Clinical Modification (ICD-9-CM)  
    Hiec, // Home Infusion EDI Coalition (HIEC)  
    Ndc, // National Drug Code in 5-4-2 format  
    MutuallyDefined  
}
```

7.3.9 WsGender

```
public enum WsGender
{
    Unknown,
    Male,
    Female
}
```

7.3.10 WsInformationStatus

```
public enum WsInformationStatus
{
    Unknown,
    Partial,
    Current,
    Latest,
    Oldest,
    Prior,
    SecondMostCurrent,
    ThirdMostCurrent
}
```

7.3.11 WsInsuranceType

```
public enum WsInsuranceType
{
    Unknown,
    Disability,
    MedicarePos,
    MedicareAdvantagePos,
    MultipleOptionsHealthPlan,
    MedicareSecondaryWorkingAgedGroupHealthPlan,
    MedicareSecondaryEndStageRenalGroupHealthPlan,
    MedicareSecondaryNofault,
    MedicareSecondaryWorkersCompensation,
    MedicareSecondaryPublicHealth,
    Dental,
    Vision,
    PrescriptionDrugs,
    MedicareSecondaryBlackLung,
    MedicareSecondaryVeteransAdministration,
    MedicareSecondaryDisabledUnder65GroupHealthPlan,
    MedicareSecondaryOtherLiability,
    AutoInsurancePolicy,
    Commercial,
    Cobra,
    MedicareConditionallyPrimary,
    DisabilityBenefits,
    ExclusiveProviderOrganization,
    FamilyOrFriends,
    GroupPolicy,
}
```

```

Hmo,
HmoMedicareRisk,
SpecialLowIncomeMedicare,
Indemnity,
IndividualPolicy,
LongTermCare,
LongTermPolicy,
LifeInsurance,
Litigation,
MedicarePartA,
MedicarePartB,
Medicaid,
MedicarePartD,
MedicareAdvantagePpo,
MedicareAdvantageFeeForService,
MedigapPartA,
MedigapPartB,
MedicareAdvantageHmo,
MedicareAdvantageHmoRisk,
MedicareAdvantageIndemnity,
MedicareAdvantagePpoPartD,
MedicareAdvantageIndemnityNoPartD,
MedicareAdvantagePosPartD,
MedicarePrimary,
MedicareAdvantageRiskHmoNoPartD,
MedicareAdvantageHmoNoPartD,
MedicareAdvantage,
Other,
PropertyInsurancePersonal,
Personal,
PersonalPayment,
Ppo,
Pos,
QualifiedMedicareBeneficiary,
PropertyInsuranceReal,
SupplementalPolicy,
Tefra,
WorkersCompensation,
WrapUpPolicy
}

```

7.3.12 WsPayeeType

```

public enum WsPayeeType
{

```

```

    Unknown,
    Provider,
    ThirdPartyAdministrator,
    Employer,
    Hospital,

```

```
    Facility,  
    GatewayProvider,  
    PlanSponsor,  
    Payer  
}
```

7.3.13 WsPayerResponseType

```
public enum WsPayerResponseType  
{  
    Unknown,  
    ClaimStatus,  
    RemittanceAdvice,  
    EmdeonClaimStatus,  
    InterchangeStatus,  
    BenefitResponse,  
}
```

7.3.14 WsPayerType

```
public enum WsPayerType  
{  
    Unknown,  
    ThirdPartyAdministrator,  
    Employer,  
    GatewayProvider,  
    PlanSponsor,  
    Payer,  
}
```

7.3.15 WsPlaceOfServiceCode

```
public enum WsPlaceOfServiceCode  
{  
    Unknown,  
    Pharmacy,  
    School,  
    HomelessShelter,  
    IhsFreestandingFacility,  
    IhsProviderFacility,  
    Tribal638FreestandingFacility,  
    Tribal638ProviderFacility,  
    CorrectionalFacility,  
    Office,  
    Home,  
    AssistedLivingFacility,  
    GroupHome,  
    MobileUnit,  
    TemporaryLodging,  
    WalkinRetailHealthClinic,  
    PlaceOfEmploymentWorksite,  
    UrgentCareFacility,  
    InpatientHospital,  
}
```

```

    OutpatientHospital,
    EmergencyRoomHospital,
    AmbulatorySurgicalCenter,
    BirthingCenter,
    MilitaryTreatmentFacility,
    SkilledNursingFacility,
    NursingFacility,
    CustodialCareFacility,
    Hospice,
    AmbulanceLand,
    AmbulanceAirOrWater,
    IndependentClinic,
    FederallyQualifiedHealthCenter,
    InpatientPsychiatricFacility,
    PsychiatricFacilityPartialHospitalization,
    CommunityMentalHealthCenter,
    IntermediateCareMentalFacility,
    ResidentialSubstanceAbuseTreatmentFacility,
    PsychiatricResidentialTreatmentCenter,
    NonresidentialSubstanceAbuseTreatmentFacility,
    MassImmunizationCenter,
    ComprehensiveInpatientRehabilitationFacility,
    ComprehensiveOutpatientRehabilitationFacility,
    EndstageRenalDiseaseTreatmentFacility,
    PublicHealthClinic,
    RuralHealthClinic,
    IndependentLaboratory,
    OtherPlaceOfService
}

```

7.3.16 WsProviderRole

```

public enum WsProviderRole
{

```

```

    Unknown,
    Hospital,
    RuralHealthClinic,
    Admitting,
    AssistantSurgeon,
    Attending,
    Billing,
    BillingService,
    Consulting,
    Covering,
    HomeHealthCare,
    Laboratory,
    OnStaff,
    Operating,
    Ordering,
    OtherPhysician,

```

```
Pharmacist,  
Pharmacy,  
PrimaryCarePhysician,  
Performing,  
PayTo,  
Purchasing,  
Referring,  
ReportingProvider,  
Submitting,  
SkilledNursingFacility,  
Supervising  
}
```

7.3.17 WsQuantityUnits

```
public enum WsQuantityUnits  
{
```

```
    Unknown,  
    Minimum,  
    AcknowledgedQuantity,  
    QuantityUsed,  
    UnacknowledgedQuantity,  
    CoveredActual,  
    CoinsuredActual,  
    CoveredEstimated,  
    NumberOfCoInsuranceDays,  
    DeductibleBloodUnits,  
    Days,  
    Units,  
    Hours,  
    LifetimeReserveActual,  
    LifetimeReserveEstimated,  
    Maximum,  
    Month,  
    NonCoveredEstimated,  
    NotReplacedBloodUnits,  
    OutlierDays,  
    NumberOfServicesOrProcedures,  
    Prescription,  
    Patients,  
    QuantityApproved,  
    QuantityDisapproved,  
    AgeHighValue,  
    AgeLowValue,  
    Visits,  
    Years,  
    FederalMedicareOrMedicaidPaymentMandateCategory1,  
    FederalMedicareOrMedicaidPaymentMandateCategory2,  
    FederalMedicareOrMedicaidPaymentMandateCategory3,  
    FederalMedicareOrMedicaidPaymentMandateCategory4,
```

```
        FederalMedicareOrMedicaidPaymentMandateCategory5
    }
```

7.3.18 WsRejectReason

```
public enum WsRejectReason
{
    Unknown,
    RequiredDataMissing,
    NotAuthorized,
    InvalidProviderIdentification,
    InvalidProviderName,
    InvalidProviderSpecialty,
    InvalidProviderPhoneNumber,
    InvalidProviderState,
    InvalidReferringProviderIdNumber,
    ProviderIneligibleForInquiries,
    ProviderNotOnFile,
    InvalidParticipantId,
    InvalidProviderAddress,
    PayerNameOrIdentifierMissing,
    InvalidPayerIdentification,
    InvalidSubscriberIdentification,
    InvalidSubscriberBirthDate,
    InvalidSubscriberName
}
```

7.3.19 WsRelationship

```
public enum WsRelationship
{
    Invalid,
    Spouse,
    Self,
    Child,
    Employee,
    RelationshipUnknown,
    OtherAdult,
    OrganDonor,
    CadaverDonor,
    LifePartner,
    OtherRelationship
}
```

7.3.20 WsRequestType

```
public enum WsRequestType
{
    Unknown,
    Request,
    SpendDownRequest,
    SpendDownCancellation
}
```

7.3.21 WsResponseServiceStatus

public enum WsResponseServiceStatus

```
{  
    Unknown,  
    ActiveReserve,  
    ActiveMilitaryOverseas,  
    AcademyStudent,  
    PresidentialAppointee,  
    ActiveMilitaryUsa,  
    Contractor,  
    DishonorablyDischarged,  
    HonorablyDischarged,  
    InactiveReserves,  
    MilitaryLeaveOfAbsence,  
    PlanToEnlist,  
    Recommissioned,  
    RetiredMilitaryOverseas,  
    RetiredWithoutRecall,  
    RetiredMilitaryUsa  
}
```

7.3.22 WsServiceAffiliation

public enum WsServiceAffiliation

```
{  
    Unknown,  
    AirForce,  
    AirForceReserves,  
    Army,  
    ArmyReserves,  
    CoastGuard,  
    MarineCorps,  
    MarineCorpsReserves,  
    NationalGuard,  
    Navy,  
    NavyReserves,  
    Other,  
    PeaceCorp,  
    RegularArmedForces,  
    Reserves,  
    USPublicHealthService,  
    ForeignMilitary,  
    AmericanRedCross,  
    DepartmentOfDefense,  
    UnitedServicesOrganization,  
    MilitarySealiftCommand  
}
```

7.3.23 WsServiceRank

public enum WsServiceRank

{

Unknown,
Admiral,
Airman,
AirmanFirstClass,
BasicAirman,
BrigadierGeneral,
Captain,
ChiefMasterSergeant,
ChiefPettyOfficer,
ChiefWarrant,
Colonel,
Commander,
Commodore,
Corporal,
CorporalSpec4,
Ensign,
FirstLieutenant,
FirstSergeant,
FirstSergeantMasterSergeant,
FleetAdmiral,
General,
GunnerySergeant,
LanceCorporal,
Lieutenant,
LieutenantColonel,
LieutenantCommander,
LieutenantGeneral,
LieutenantJuniorGrade,
Major,
MajorGeneral,
MasterChiefPettyOfficer,
MasterGunnerySergeantMajor,
MasterSergeant,
MasterSergeantSpec8,
PettyOfficerFirstClass,
PettyOfficerSecondClass,
PettyOfficerThirdClass,
Private,
PrivateFirstClass,
RearAdmiral,
Recruit,
Seaman,
SeamanApprentice,
SeamanRecruit,
SecondLieutenant,
SeniorChiefPettyOfficer,
SeniorMasterSergeant,

```
Sergeant,  
SergeantFirstClassSpec7,  
SergeantMajorSpec9,  
SergeantSpec5,  
StaffSergeant,  
StaffSergeantSpec6,  
TechnicalSergeant,  
ViceAdmiral,  
WarrantOfficer  
}
```

7.3.24 WsServiceType

```
public enum WsServiceType  
{  
    Unknown,  
    MedicalCare,  
    Surgical,  
    Consultation,  
    DiagnosticXRay,  
    DiagnosticLab,  
    RadiationTherapy,  
    Anesthesia,  
    SurgicalAssistance,  
    OtherMedical,  
    NonEscrowOrNonImpoundService,  
    EscrowOrImpoundService,  
    BloodCharges,  
    UsedDurableMedicalEquipment,  
    DurableMedicalEquipmentPurchase,  
    AmbulatoryServiceCenterFacility,  
    RenalSuppliesInTheHome,  
    AlternateMethodDialysis,  
    ChronicRenalDiseaseCrdEquipment,  
    PreAdmissionTesting,  
    DurableMedicalEquipmentRental,  
    PneumoniaVaccine,  
    SecondSurgicalOpinion,  
    ThirdSurgicalOpinion,  
    SocialWork,  
    DiagnosticDental,  
    Periodontics,  
    Restorative,  
    Endodontics,  
    MaxillofacialProsthetics,  
    AdjunctiveDentalServices,  
    HealthBenefitPlanCoverage, // the most general value  
    BenefitDisclaimer,  
    PlanWaitingPeriod,  
    Chiropractic,  
}
```

ChiropracticOfficeVisits,
DentalCare,
DentalCrowns,
DentalAccident,
Orthodontics,
Prosthodontics,
OralSurgery,
RoutinePreventiveDental,
HomeHealthCare,
HomeHealthPrescriptions,
HomeHealthVisits,
Hospice,
RespiteCare,
Hospital,
HospitalInpatient,
HospitalRoomAndBoard,
HospitalOutpatient,
HospitalEmergencyAccident,
HospitalEmergencyMedical,
HospitalAmbulatorySurgical,
LongTermCare,
MajorMedical,
MedicallyRelatedTransportation,
AirTransportation,
Cabulance,
LicensedAmbulance,
GeneralBenefits,
InVitroFertilization,
MriCatScan,
DonorProcedures,
Acupuncture,
NewbornCare,
Pathology,
SmokingCessation,
WellBabyCare,
Maternity,
Transplants,
AudiologyExam,
InhalationTherapy,
DiagnosticMedical,
PrivateDutyNursing,
ProstheticDevice,
Dialysis,
OtologicalExam,
Chemotherapy,
AllergyTesting,
Immunizations,
RoutinePhysical,

FamilyPlanning,
Infertility,
Abortion,
Aids,
EmergencyServices,
Cancer,
Pharmacy,
FreeStandingPrescriptionDrug,
MailOrderPrescriptionDrug,
BrandNamePrescriptionDrug,
GenericPrescriptionDrug,
Podiatry,
PodiatryOfficeVisits,
PodiatryNursingHomeVisits,
ProfessionalPhysician,
Anesthesiologist,
ProfessionalPhysicianVisitOffice,
ProfessionalPhysicianVisitInpatient,
ProfessionalPhysicianVisitOutpatient,
ProfessionalPhysicianVisitNursingHome,
ProfessionalPhysicianVisitSkilledNursingFacility,
ProfessionalPhysicianVisitHome,
Psychiatric,
PsychiatricRoomAndBoard,
Psychotherapy,
PsychiatricInpatient,
PsychiatricOutpatient,
Rehabilitation,
RehabilitationRoomAndBoard,
RehabilitationInpatient,
RehabilitationOutpatient,
OccupationalTherapy,
PhysicalMedicine,
SpeechTherapy,
SkilledNursingCare,
SkilledNursingCareRoomAndBoard,
SubstanceAbuse,
Alcoholism,
DrugAddiction,
VisionOptometry,
Frames,
RoutineExam,
Lenses,
RoutineEyeExam,
NonmedicallyNecessaryPhysical,
ExperimentalDrugTherapy,
BurnCare,
BrandNamePrescriptionDrugFormulary,

BrandNamePrescriptionDrugNonFormulary,
IndependentMedicalEvaluation,
PartialHospitalizationPsychiatric,
DayCarePsychiatric,
CognitiveTherapy,
MassageTherapy,
PulmonaryRehabilitation,
CardiacRehabilitation,
Pediatric,
Nursery,
Skin,
Orthopedic,
Cardiac,
Lymphatic,
Gastrointestinal,
Endocrine,
Neurology,
Eye,
InvasiveProcedures,
Gynecological,
Obstetrical,
ObstetricalGynecological,
MailOrderPrescriptionDrugBrandName,
MailOrderPrescriptionDrugGeneric,
PhysicianVisitOfficeSick,
PhysicianVisitOfficeWell,
CoronaryCare,
PrivateDutyNursingInpatient,
PrivateDutyNursingHome,
SurgicalBenefitsProfessionalPhysician,
SurgicalBenefitsFacility,
MentalHealthProviderInpatient,
MentalHealthProviderOutpatient,
MentalHealthFacilityInpatient,
MentalHealthFacilityOutpatient,
SubstanceAbuseFacilityInpatient,
SubstanceAbuseFacilityOutpatient,
ScreeningXRay,
ScreeningLaboratory,
MammogramHighRiskPatient,
MammogramLowRiskPatient,
FluVaccination,
EyewearAndEyewearAccessories,
CaseManagement,
Dermatology,
DurableMedicalEquipment,
DiabeticSupplies,
GenericPrescriptionDrugFormulary,

```
GenericPrescriptionDrugNonFormulary,  
Allergy,  
IntensiveCare,  
MentalHealth,  
NeonatalIntensiveCare,  
Oncology,  
PositronEmissionTomographyPetScan,  
PhysicalTherapy,  
Pulmonary,  
Renal,  
ResidentialPsychiatricTreatment,  
TransitionalCare,  
TransitionalNurseryCare,  
UrgentCare,  
}
```

7.3.25 WsTimePeriod

```
public enum WsTimePeriod
```

```
{  
    Unknown,  
    Hour,  
    Day,  
    TwentyFourHours,  
    Years,  
    ServiceYear,  
    CalendarYear,  
    YearToDate,  
    Contract,  
    Episode,  
    Visit,  
    Outlier,  
    Remaining,  
    Exceeded,  
    NotExceeded,  
    Lifetime,  
    LifetimeRemaining,  
    Month,  
    Week,  
    Admission  
}
```

7.3.26 WsYesOrNo

```
public enum WsYesOrNo
```

```
{  
    Unknown,  
    No,  
    NotApplicable,  
    Yes  
}
```

7.4 Classes

7.4.1 WsAdditionalBenefitInformation

```
public class WsAdditionalBenefitInformation
{
    public string NcciNatureOfInjuryCode { get; set; }
    public string OtherNatureOfInjuryCode { get; set; }
    public WsPlaceOfServiceCode? PlaceOfService { get; set; }
    public string InjuredBodyPart { get; set; }
}
```

7.4.2 WsAddress

```
public class WsAddress
{
    public string SendToAttentionOf { get; set; }
    public string Address1 { get; set; }
    public string Address2 { get; set; }
    public string City { get; set; }
    public string State { get; set; }
    public string ZipCode { get; set; }
    public string CountryCode { get; set; }
    public string CountrySubCode { get; set; }
}
```

7.4.3 WsBenefit

```
public class WsBenefit
{
    public List<WsServiceType> Type { get; set; }
    public WsPlaceOfServiceCode? PlaceOfService { get; set; }
    public string ReferralNumber { get; set; }
    public string PriorAuthorizationNumber { get; set; }
    public WsDateRange PlanDate { get; set; }
    public bool IsFamilyBenefit { get; set; }
    public WsEligibilityRequestProcedure Procedure { get; set; }
}
```

7.4.4 WsBenefitEntity

```
public class WsBenefitEntity
{
    public WsEntityKind Kind { get; set; }
    public WsEntityType Type { get; set; }
    public string CommonName { get; set; }
    public string FirstName { get; set; }
    public string MiddleName { get; set; }
    public string Suffix { get; set; }
    public WsAddress PhysicalAddress { get; set; }
    public string DodHealthServiceRegion { get; set; }
    public List<WsResponseContact> Contacts { get; set; }
    public string TaxonomyCode { get; set; }
    public WsProviderRole? TaxonomyRole { get; set; }
    public string BenefitRelatedId { get; set; }
}
```

```

    public string NationalProviderId { get; set; }
    public string EmployersIdNumber { get; set; }
    public string TaxIdNumber { get; set; }
    public string CmsPlanId { get; set; }
    public string SocialSecurityNumber { get; set; }
    public string PharmacyNumber { get; set; }
    public string PayerId { get; set; }
    public string StandardUniqueHealthId { get; set; }
    public string Etin { get; set; }
    public string FacilityId { get; set; }
    public string PriorIdNumber { get; set; }
    public string FacilityIdNumber { get; set; }
    public string ServiceProviderNumber { get; set; }
    public string NaicId { get; set; }
    public string MemberIdNumber { get; set; }
    public WsRelationship? BenefitRelationshipToPatient { get; set; }
}

```

7.4.5 WsBenefitInformation

```

public class WsBenefitInformation
{
    public WsBenefitInformationType Type { get; set; }
    public List<WsServiceType> TypesOfService { get; set; }
    public WsCoverageLevel? LevelOfCoverage { get; set; }
    public WsPlaceOfServiceCode? PlaceOfService { get; set; }
    public WsInsuranceType? TypeOfInsurance { get; set; }
    public string PlanCoverageDescription { get; set; }
    public WsTimePeriod? BenefitTimePeriod { get; set; }
    public decimal ? BenefitAmount { get; set; }
    public decimal ? PatientResponsibilityPercentage { get; set; }
    public WsQuantity? BenefitQuantity { get; set; }
    public WsYesOrNo? AuthenticationOrCertificationRequired { get; set; }
    public WsYesOrNo? PlanInNetwork { get; set; }
    public string ReferralNumber { get; set; }
    public string PriorAuthorizationNumber { get; set; }
    public WsEligibilityResponseProcedure Procedure { get; set; }
    public List<WsServiceDeliverySchedule> DeliverySchedules { get; set; }
    public string PlanNumber { get; set; }
    public string GroupOrPolicyNumber { get; set; }
    public string MemberId { get; set; }
    public string MemberIdSuffix { get; set; }
    public string GroupNumber { get; set; }
    public string PolicyNumber { get; set; }
    public string AlternativeListId { get; set; }
    public string CoverageListId { get; set; }
    public string DrugFormularyNumber { get; set; }
    public string MedicaidAssistanceCategory { get; set; }
    public string MedicaidRecipientId { get; set; }
    public string HealthInsuranceClaimNumber { get; set; }
}

```



```

    public string PlanNetworkIdNumber { get; set; }
    public string PlanOrGroupName { get; set; }
    public DateTime? DischargeDate { get; set; }
    public WsDateRange PeriodDates { get; set; }
    public DateTime? CompletionDate { get; set; }
    public DateTime? CobDate { get; set; }
    public DateTime? PlanDate { get; set; }
    public WsDateRange BenefitDates { get; set; }
    public DateTime? PcpDate { get; set; }
    public DateTime? LatestVisitDate { get; set; }
    public WsDateRange EligibilityDates { get; set; }
    public DateTime? DateAdded { get; set; }
    public DateTime? PlanBeginDate { get; set; }
    public DateTime? AdmissionDate { get; set; }
    public DateTime? ServiceDate { get; set; }
    public DateTime? LastUpdateDate { get; set; }
    public DateTime? StatusDate { get; set; }
    public List<WsRequestError> Errors { get; set; }
    public List<string> Messages { get; set; }
    public List<WsAdditionalBenefitInformation> AdditionalInformation { get; set; }
    public List<WsBenefitEntity> BenefitEntities { get; set; }
}

```

7.4.6 WsBenefitInquiry

```

public class WsBenefitInquiry : WsDocument
{
    public WsRequestType RequestType { get; set; }
    public string OriginalSubmitterTransactionId { get; set; }
    public string ApexSubmitterTransactionId { get; set; }
    public List<WsEligibilityRequestPayer> Payers { get; set; }
}

```

7.4.7 WsBenefitResponse

```

public class WsBenefitResponse : WsPayerResponse
{
    public string GroupControlNumber { get; set; }
    public string TradingPartnerId { get; set; }
    public List<WsEligibilityResponsePayer> Payers { get; set; }
    public List<WsRequestError> EnvelopeErrors { get; set; }
}

```

7.4.8 WsBenefitResponseResult

Same members as WsBenefitSubmitResult.

7.4.9 WsBenefitSubmitResult

```

public class WsBenefitSubmitResult
{
    public string RequestId { get; set; }
    public WsBenefitStatus Status { get; set; }
    public WsBenefitResponse[] Responses { get; set; }
    public WsValidationResult[] ValidationResults { get; set; }
}

```

```
        public WsErrorResponse[] ErrorResponses { get; set; }
    }
}
```

7.4.10 WsDateRange

```
public class WsDateRange
{
    public DateTime? Start { get; set; }
    public DateTime? End { get; set; }
}
```

7.4.11 WsDiagnosisCode

```
public class WsDiagnosisCode
{
    public WsDiagnosticCodeType Type { get; set; }
    public string Code { get; set; }
}
```

7.4.12 WsEligibilityRequestDependent

```
public class WsEligibilityRequestDependent : WsEligibilityRequestPatient
{
    public WsBenefit RequestedBenefits { get; set; }
    public WsRelationship RelationshipToSubscriber { get; set; }
}
```

7.4.13 WsEligibilityRequestPatient

```
public class WsEligibilityRequestPatient
{
    public WsEligibilityRequestProvider EligibilityRequestProvider { get; set; }
    public WsTraceNumber PayeeTraceNumber { get; set; }
    public string CommonName { get; set; }
    public string FirstName { get; set; }
    public string MiddleName { get; set; }
    public string Suffix { get; set; }
    public WsAddress PhysicalAddress { get; set; }
    public int ? MultipleBirthSequenceNumber { get; set; }
    public DateTime? BirthDate { get; set; }
    public WsGender? Gender { get; set; }
    public WsDateRange IssueDate { get; set; }
    public WsDateRange PlanDate { get; set; }
    public string PlanNumber { get; set; }
    public string MemberId { get; set; }
    public string StandardUniqueHealthId { get; set; }
    public string PatientAccountNumber { get; set; }
    public string GroupOrPolicyNumber { get; set; }
    public string GroupNumber { get; set; }
    public string PolicyNumber { get; set; }
    public string ContractNumber { get; set; }
    public string CaseNumber { get; set; }
    public string InsurancePolicyNumber { get; set; }
    public string MedicalRecordNumber { get; set; }
    public string HealthInsuranceClaimNumber { get; set; }
}
```

```

    public string MedicaidRecipientId { get; set; }
    public string IdCardSerialNumber { get; set; }
    public string IdCardNumber { get; set; }
    public string IssueNumber { get; set; }
    public string EligibilityCategory { get; set; }
    public string PlanNetworkIdNumber { get; set; }
    public string SocialSecurityNumber { get; set; }
    public string AgencyClaimNumber { get; set; }
}

```

7.4.14 WsEligibilityRequestPayee

```

public class WsEligibilityRequestPayee
{
    public WsPayeeType Type { get; set; }
    public WsEntityType EntityType { get; set; }
    public WsProviderRole ProviderRole { get; set; }
    public string CommonName { get; set; }
    public string FirstName { get; set; }
    public string MiddleName { get; set; }
    public string Suffix { get; set; }
    public string NationalProviderId { get; set; }
    public string EmployersIdNumber { get; set; }
    public string TaxIdNumber { get; set; }
    public string CmsPlanId { get; set; }
    public string SocialSecurityNumber { get; set; }
    public string PharmacyNumber { get; set; }
    public string PayerId { get; set; }
    public string StateLicenseNumber { get; set; }
    public string StateIssuingLicense { get; set; }
    public string MedicareProviderNumber { get; set; }
    public string MedicaidProviderNumber { get; set; }
    public string PriorIdNumber { get; set; }
    public string FacilityIdNumber { get; set; }
    public string FacilityNetworkIdNumber { get; set; }
    public string SubmitterIdNumber { get; set; }
    public string ContractNumber { get; set; }
    public string PersonalIdNumber { get; set; }
    public string ServiceProviderNumber { get; set; }
    public string DevicePin { get; set; }
    public string UserId { get; set; }
    public string RenderingProviderTaxonomyCode { get; set; }
    public string ReceivingProviderTaxonomyCode { get; set; }
    public WsAddress PhysicalAddress { get; set; }
    public List<WsEligibilityRequestSubscriber> Subscribers { get; set; }
}

```

7.4.15 WsEligibilityRequestPayer

```

public class WsEligibilityRequestPayer
{

```

```

    public WsPayerType Type { get; set; }
    public WsEntityType EntityType { get; set; }
    public string VendorPayerId { get; set; }
    public string CommonName { get; set; }
    public string FirstName { get; set; }
    public string MiddleName { get; set; }
    public string Suffix { get; set; }
    public string PayerId { get; set; }
    public string CmsPlanId { get; set; }
    public string EmployersIdNumber { get; set; }
    public string NationalProviderId { get; set; }
    public string ElectronicTransmitterIdNumber { get; set; }
    public string NationalPayerIdNumber { get; set; }
    public string TaxIdNumber { get; set; }
    public string NaicId { get; set; }
    public List<WsEligibilityRequestPayee> Payees { get; set; }
}

```

7.4.16 WsEligibilityRequestProcedure

```

public class WsEligibilityRequestProcedure
{
    public WsEligibilityProcedureCodeSource Type { get; set; }
    public string ProcedureCode { get; set; }
    public List<string> Modifiers { get; set; }
    public List<WsDiagnosisCode> DiagnosisCodes { get; set; }
}

```

7.4.17 WsEligibilityRequestSubscriber

```

public class WsEligibilityRequestSubscriber : WsEligibilityRequestPatient
{
    public WsEntityType? EntityType { get; set; }
    public List<WsSubscriberBenefit> RequestedBenefits { get; set; }
    public List<WsEligibilityRequestDependent> Dependents { get; set; }
}

```

7.4.18 WsEligibilityResponsePatient

```

public class WsEligibilityResponsePatient
{
    public WsTraceNumber ClearinghouseTraceNumber { get; set; }
    public WsTraceNumber PayeeTraceNumber { get; set; }
    public WsTraceNumber PayerTraceNumber { get; set; }
    public bool IsInsured { get; set; }
    public string CommonName { get; set; }
    public string FirstName { get; set; }
    public string MiddleName { get; set; }
    public string Suffix { get; set; }
    public WsAddress PhysicalAddress { get; set; }
    public DateTime? BirthDate { get; set; }
    public WsGender Gender { get; set; }
    public List<WsBenefitInformation> Benefits { get; set; }
}

```

```

public DateTime? DischargeDate { get; set; }
public WsDateRange IssueDate { get; set; }
public DateTime? EffectiveDateOfChange { get; set; }
public WsDateRange PlanDate { get; set; }
public DateTime? EligibilityDate { get; set; }
public DateTime? DateAdded { get; set; }
public WsDateRange CobraDates { get; set; }
public WsDateRange PremiumPaidDates { get; set; }
public WsDateRange PlanDates { get; set; }
public WsDateRange EligibilityDates { get; set; }
public DateTime? EnrollmentDate { get; set; }
public DateTime? AdmissionDate { get; set; }
public DateTime? DeathDate { get; set; }
public DateTime? CertificationDate { get; set; }
public DateTime? ServiceDate { get; set; }
public WsDateRange PolicyDates { get; set; }
public DateTime? DateOfLastUpdate { get; set; }
public DateTime? StatusDate { get; set; }
public string MemberId { get; set; }
public string MemberIdSuffix { get; set; }
public string OriginalPrimaryId { get; set; }
public string PlanOrGroupName { get; set; }
public string StandardUniquelId { get; set; }
public string PlanNumber { get; set; }
public string PatientAccountNumber { get; set; }
public string GroupOrPolicyNumber { get; set; }
public string GroupNumber { get; set; }
public string PolicyNumber { get; set; }
public string ClassOfContract { get; set; }
public string ContractNumber { get; set; }
public string MedicalRecordNumber { get; set; }
public string HealthInsuranceClaimNumber { get; set; }
public string IdCardSerialNumber { get; set; }
public string IdentityCardNumber { get; set; }
public string IssueNumber { get; set; }
public string EligibilityCategory { get; set; }
public string PlanNetworkIdNumber { get; set; }
public string SocialSecurityNumber { get; set; }
public string AgencyClaimNumber { get; set; }
public string MedicaidRecipientId { get; set; }
public string TaxonomyCode { get; set; }
public string FamilyUnitNumber { get; set; }
public WsProviderRole? TaxonomyRole { get; set; }
public int ? MultipleBirthSequenceNumber { get; set; }
public bool PrimaryInformationChanged { get; set; }
public WsMilitaryService MilitaryService { get; set; }
public List<WsRequestError> Errors { get; set; }
}

```

7.4.19 WsEligibilityResponsePayee

```
public class WsEligibilityResponsePayee
{
    public WsPayeeType? Type { get; set; }
    public WsEntityType? EntityType { get; set; }
    public WsProviderRole? ProviderRole { get; set; }
    public string CommonName { get; set; }
    public string FirstName { get; set; }
    public string MiddleName { get; set; }
    public string Suffix { get; set; }
    public string NationalProviderId { get; set; }
    public string EmployersIdNumber { get; set; }
    public string TaxIdNumber { get; set; }
    public string CmsPlanId { get; set; }
    public string SocialSecurityNumber { get; set; }
    public string PharmacyNumber { get; set; }
    public string PayerId { get; set; }
    public string ServiceProviderNumber { get; set; }
    public string StateLicenseNumber { get; set; }
    public string StateIssuingLicense { get; set; }
    public string MedicareProviderNumber { get; set; }
    public string MedicaidProviderNumber { get; set; }
    public string PriorIdNumber { get; set; }
    public string PersonalIdNumber { get; set; }
    public string FacilityIdNumber { get; set; }
    public string FacilityNetworkIdNumber { get; set; }
    public string SubmitterIdNumber { get; set; }
    public string ContractNumber { get; set; }
    public string DevicePin { get; set; }
    public string ProviderPlanNetworkId { get; set; }
    public string UserId { get; set; }
    public string ReceivingProviderTaxonomyCode { get; set; }
    public WsAddress PhysicalAddress { get; set; }
    public List<WsRequestError> Errors { get; set; }
    public List<WsEligibilityResponseSubscriber> Subscribers { get; set; }
}
```

7.4.20 WsEligibilityResponsePayer

```
public class WsEligibilityResponsePayer
{
    public WsPayerType? Type { get; set; }
    public WsEntityType? EntityType { get; set; }
    public string CommonName { get; set; }
    public string FirstName { get; set; }
    public string MiddleName { get; set; }
    public string Suffix { get; set; }
    public string PayerId { get; set; }
    public string CmsPlanId { get; set; }
}
```

```

    public string EmployersIdNumber { get; set; }
    public string NationalProviderId { get; set; }
    public string ElectronicTransmitterIdNumber { get; set; }
    public string NationalPayerIdNumber { get; set; }
    public string TaxIdNumber { get; set; }
    public string NaicId { get; set; }
    public List<WsEligibilityResponsePayee> Payees { get; set; }
    public List<WsResponseContact> Contacts { get; set; }
    public List<WsRequestError> Errors { get; set; }
}

```

7.4.21 WsEligibilityResponseProcedure

```

public class WsEligibilityResponseProcedure
{
    public WsEligibilityProcedureCodeSource Type { get; set; }
    public string StartingProcedureCode { get; set; }
    public string EndingProcedureCode { get; set; }
    public List<string> Modifiers { get; set; }
    public List<WsDiagnosisCode> DiagnosisCodes { get; set; }
}

```

7.4.22 WsEligibilityResponseSubscriber

```

public class WsEligibilityResponseSubscriber : WsEligibilityResponsePatient
{
    public WsEntityType? EntityType { get; set; }
    public string CaseNumber { get; set; }
    public List<WsEligibilityResponseDependent> Dependents { get; set; }
}

```

7.4.23 WsErrorResponse

```

public class WsErrorResponse
{
    public WsTraceNumber ClearinghouseTraceNumber { get; set; }
    public WsTraceNumber PayeeTraceNumber { get; set; }
    public string ErrorMessage { get; set; }
}

```

7.4.24 WsMilitaryService

```

public class WsMilitaryService
{
    public WsInformationStatus Status { get; set; }
    public WsResponseServiceStatus StatusOfService { get; set; }
    public WsServiceAffiliation Affiliation { get; set; }
    public WsServiceRank? Rank { get; set; }
    public string Description { get; set; }
    public WsDateRange ServiceDate { get; set; }
}

```

7.4.25 WsPayerResponse

```

public class WsPayerResponse : WsDocument
{
    public string DocumentId { get; set; }
    public DateTime? CreateDate { get; set; }
}

```

```
        public WsPayerResponseType ResponseType { get; set; }  
    }
```

7.4.26 WsQuantity

```
public struct WsQuantity  
{  
    public WsQuantityUnits Unit { get; set; }  
    public decimal Amount { get; set; }  
}
```

7.4.27 WsRequestError

```
public class WsRequestError  
{  
    public bool InvalidRequest { get; set; }  
    public WsRejectReason Reason { get; set; }  
    public WsFollowupAction Action { get; set; }  
    public string PayerId { get; set; }  
    public string Message { get; set; }  
    public string ErrorCode { get; set; }  
}
```

7.4.28 WsResponseContact

```
public class WsResponseContact  
{  
    public string Name { get; set; }  
    public List<WsResponseContactMethods> Methods { get; set; }  
}
```

7.4.29 WsResponseContactMethods

```
public class WsResponseContactMethods  
{  
    public string EmailAddress { get; set; }  
    public string PhoneNumber { get; set; }  
    public string WorkPhoneNumber { get; set; }  
    public string PhoneNumberExtension { get; set; }  
    public string FaxNumber { get; set; }  
    public string Url { get; set; }  
    public string EdiAccessNumber { get; set; }  
}
```

7.4.30 WsServiceDeliverySchedule

```
public class WsServiceDeliverySchedule  
{  
    public WsQuantity? Quality { get; set; }  
    public WsSampleUnits? SampleFrequencyUnit { get; set; }  
    public decimal? SampleFrequency { get; set; }  
    public WsTimePeriod? TimePeriod { get; set; }  
    public decimal? PeriodCount { get; set; }  
    public WsDeliveryFrequency? DeliveryFrequency { get; set; }  
    public WsDeliveryTime? DeliveryTime { get; set; }  
}
```


7.4.31 WsSubscriberBenefit

```
public class WsSubscriberBenefit : WsBenefit
{
    public decimal? SpendDownAmount { get; set; }
    public decimal? SpendDownTotalBilledAmount { get; set; }
}
```

7.4.32 WsTraceNumber

See the section on identifying patients.

```
public class WsTraceNumber
{
    public string OriginatorId { get; set; }
    public string OriginatorSecondaryId { get; set; }
    public string Number { get; set; }
}
```

7.4.33 WsValidationBase

```
public class WsValidationBase
{
    public List<WsRequestError> Errors { get; set; }
    public Dictionary<string, string> Ids { get; set; }
}
```

7.4.34 WsValidationDependent

```
public class WsValidationDependent : WsValidationPatient
{
}
```

7.4.35 WsValidationPatient

```
public class WsValidationPatient : WsValidationBase
{
}
```

7.4.36 WsValidationPayee

```
public class WsValidationPayee : WsValidationBase
{
    public List<WsValidationSubscriber> Subscribers { get; set; }
}
```

7.4.37 WsValidationPayer

```
public class WsValidationPayer : WsValidationBase
{
    public List<WsValidationPayee> Payees { get; set; }
}
```

7.4.38 WsValidationResult

```
public class WsValidationResult
{
    public string OriginalSubmitterTransactionId { get; set; }
    public List<WsValidationPayer> ValidationPayers { get; set; }
    public List<WsRequestError> GeneralErrors { get; set; }
}
```

7.4.39 WsValidationSubscriber

```
public class WsValidationSubscriber : WsValidationPatient
{
    public List<WsValidationDependent> Dependents { get; set; }
}
```

8 Sample Code

8.1 Submitting an Eligibility Request

8.1.1 URLs

Apex OneTouch Service

<https://sandbox.services.apexedi.com/v1/onetouch.soap>

<https://production.services.apexedi.com/v1/onetouch.soap>

8.1.2 Disclaimer

This information contains code samples in source language, which illustrate potential uses of the ApexEDI APIs. You may copy, modify, and distribute these samples in any form without payment to ApexEDI, for the purposes of implementing and using the ApexEDI APIs. These examples have not been thoroughly tested, so ApexEDI cannot guarantee or imply their reliability, serviceability, or function. The samples are provided "AS IS", without warranty of any kind. ApexEDI shall not be liable for any damages arising out of your use of the code samples.

8.1.3 Setup Instructions – Visual Studio:

For this sample code, create a new .NET Console project in your solution, to match the code below, give it the name EligibilityDocSampleSource. Under References, add a service reference using the service reference path of the Apex Sandbox URL:

<https://sandbox.services.apexedi.com/v1/onetouch.soap>

For the NAME of the service reference, just give it OneTouchService. Choose OK, and after it has been created as a reference in your project, modify the app.config as shown below under 8.2 Client Config File

8.2 Client Config File (app.config)

NOTE: The **binding** tag should increase the sizes timeouts for larger payloads

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="BasicHttpBinding_IOneTouchServices"
          allowCookies="true"
          maxReceivedMessageSize="20000000"
          maxBufferSize="20000000"
          maxBufferPoolSize="20000000"
          closeTimeout="00:10:00"
          openTimeout="00:10:00" />
      </basicHttpBinding>
    </bindings>
  </system.serviceModel>
</configuration>
```

```

        receiveTimeout="00:10:00"
        sendTimeout="00:10:00">
    <security mode="Transport" />
    <readerQuotas maxDepth="32"
        maxLength="200000000"
        maxStringContentLength="200000000"/>
    </binding>
    <binding name="BasicHttpBinding_IOneTouchServices1" />
</basicHttpBinding>
</bindings>
<client>
    <endpoint
address=https://sandbox.services.apexedi.com/V1/onetouch.soap/soapService
        binding="basicHttpBinding"
        bindingConfiguration="BasicHttpBinding_IOneTouchServices"
        contract="OneTouchService.IOneTouchServices"
        name="BasicHttpBinding_IOneTouchServices" />
    </client>
</system.serviceModel>
</configuration>

```

8.3 C# Source Code – NOTE: DefaultVendorKey, DefaultPassword, and SiteId are all invalid in this source sample – they must be replaced with the key, password, and SiteId that you received from our Vendor Relations manager.

```

using System;
using EligibilityDocSampleSource.OneTouchService;

namespace EligibilityDocSampleSource
{
    class Program
    {
        private static string VendorKey { get; set; }
        private static string Password { get; set; }
        private static string SiteId { get; set; }

        //private const string MemberId = "1001"; // envelope errors
        //private const string MemberId = "1002"; // payer errors
        //private const string MemberId = "1003"; // payee errors
        //private const string MemberId = "1004"; // subscriber errors
    }
}

```

```

//private const string MemberId = "1005"; // dependent errors
//private const string MemberId = "1006"; // syntax error

private const string MemberId = "1007"; // complex benefits will be returned

//private const string MemberId = "1008"; // simple benefits will be returned
// after a delay of approximately 65 seconds. Note that there
// will be 65-second delay for each patient (subscriber or dependent)
// when 1008 is the subscriber member ID
//private const string MemberId = "99887766";
// (or any other member ID) simple benefits will be returned

private const string DefaultVendorKey = "TY0W3N04KSGKRQCPQ2J514UD";
private const string DefaultPassword = "BSXQZRCJC020AZ5GI4F1B0AE";

static void Main(string[] args)
{
    VendorKey = DefaultVendorKey;
    Password = DefaultPassword;
    SiteId = "ZZZ";
    SubmitEligibilitySimpleBenefits();
}

public static void SubmitEligibilitySimpleBenefits()
{
    WsBenefitSubmitRequest request =
        CreateSimpleBenefitsRequest(MemberId);
    WsBenefitSubmitResult result = SubmitEligibilityRequest(request);
    OneTouchServicesClient client = new OneTouchServicesClient();

    WsBenefitResponseResult responseResult =
        client.GetEligibilityResponses(VendorKey, Password, SiteId,
            result.RequestId);
}

private static WsBenefitSubmitRequest
    CreateSimpleBenefitsRequest(string memberId)
{
    WsEligibilityRequestPayee payee = null;
    WsEligibilityRequestSubscriber subscriber = null;
    WsEligibilityRequestPayer payer = null;
    WsBenefitSubmitRequest request = new WsBenefitSubmitRequest();
    request.BenefitInquiries =
        CreateInquiries(CreateSimpleSubscriberBenefits(),
            out payer, out payee, out subscriber, memberId);
}

```

```

        request.IsTest = true;
        return request;
    }

    private static WsSubscriberBenefit[] CreateSimpleSubscriberBenefits()
    {
        WsSubscriberBenefit benefit = new WsSubscriberBenefit();
        benefit.Type = new WsServiceType[]
            {WsServiceType.HealthBenefitPlanCoverage };
        WsSubscriberBenefit[] rval = new WsSubscriberBenefit[] { benefit };
        return rval;
    }

    private static WsBenefitInquiry[]
        CreateInquiries(WsSubscriberBenefit[] benefits,
            out WsEligibilityRequestPayer payer, out WsEligibilityRequestPayee payee,
            out WsEligibilityRequestSubscriber subscriber, string memberId)
    {
        payer = CreatePayer(benefits, out payee, out subscriber, memberId);
        WsBenefitInquiry inquiry = new WsBenefitInquiry();
        inquiry.Payers = new WsEligibilityRequestPayer[] { payer };
        inquiry.RequestType = WsRequestType.Request;
        WsBenefitInquiry[] inquiries = new WsBenefitInquiry[] { inquiry };
        return inquiries;
    }

    private static WsEligibilityRequestPayer CreatePayer(
        WsSubscriberBenefit[] benefits, out WsEligibilityRequestPayee payee,
        out WsEligibilityRequestSubscriber subscriber, string memberId)
    {
        WsTraceNumber traceNumber = new WsTraceNumber();
        traceNumber.Number = "12345678";
        traceNumber.OriginatorId = "9VENDOR999";
        subscriber = new WsEligibilityRequestSubscriber();
        subscriber.EntityType = WsEntityType.Individual;
        subscriber.MemberId = memberId;
        subscriber.CommonName = "Smith";
        subscriber.FirstName = "John";
        subscriber.BirthDate = new DateTime(1985, 1, 1);
        subscriber.PayeeTraceNumber = traceNumber;
        subscriber.RequestedBenefits = benefits;

        payee = new WsEligibilityRequestPayee();
        payee.Type = WsPayeeType.Provider;
        payee.EntityType = WsEntityType.Individual;
    }

```

```

        payee.NationalProviderId = "123456789";
        payee.CommonName = "Doe";
        payee.FirstName = "Jane";
        payee.Subscribers = new WsEligibilityRequestSubscriber[] { subscriber };

        WsEligibilityRequestPayer payer = new WsEligibilityRequestPayer();
        payer.Type = WsPayerType.Payer;
        payer.PayerId = "APX14114";
        payer.EntityType = WsEntityType.Organization;
        payer.CommonName = "Apex Test Payer";
        payer.Payees = new WsEligibilityRequestPayee[] { payee };

        return payer;
    }

    private static WsBenefitSubmitResult SubmitEligibilityRequest(
        WsBenefitSubmitRequest request)
    {
        OneTouchServicesClient oneTouchServicesClient =
            new OneTouchServicesClient();

        WsBenefitSubmitResult result =
            oneTouchServicesClient.SubmitEligibilityRequest(VendorKey,
                Password, SiteId, request);

        return result;
    }
}

```