Unit 3, The Paddock
Hambridge Road
Newbury
Berkshire
United Kingdom

Project: DBV-OSI II

Project: ONE – OPAC Network in Europe

-Z39.50 API-Definition Manual

Document No.: D-008
Part B - Data and Structure Definitions

Document Version: 7.0

Z39.50 API Software Version 2.1

Prepared by:	R. A. Bull	Crossnet
Based on Original by:	B. Hergeth	Danet GmbH

This document is based upon the DBV OSI II API definition Manual d-008 Version 5, prepared by B. Hergeth, DANET GmbH for the DBV OSI II Project.

Extensions to the DBV OSI II API have been added under the auspices of the ONE project. These extensions are included in this document and follow the style of the original design.

Table of Contents

1. Introduction

1.1 Scope

This document is the API (Application Programmers Interface) manual for the DBV OSI II API software. The software is a Freeware package for the UNIX operating system that allows developers to write Z39.50 client and server applications.

The API software was been originally developed for the German Library Project, DBV OSI II, and has been subsequently extended in the scope of the European Project ONE – OPAC Network in Europe. The API software has been developed by Crossnet Systems Limited.

This document is provided in two parts:

Part A provides a general description of the API and a description of the API functions.

Part B provides information on the API data structures.

This document supports release 2.1 of the API software. The main additions to the API for this version are the support for the Character Set and Language Negotiation record, and the inclusion of memory free functions to free memory allocated for the API data structures.

1.2 Header Files

The data definitions and structure definitions of the API which are defined in the following header files:

sidefs.h	- DBV OSI II standard definitions
sitypes.h	- DBV OSI II PDU structure type definitions
sitypes2.h	- DBV OSI II additional structure definitions
itemordr.h	- DBV OSI II item order extended service and ILL structure definition
iord_ext.h	- DBV OSI II SUBITO additional information structure definition
dbvproto.h	- prototypes of the API functions and definitions of errors.

Notes:

In order to include the API library functions in origin and target applications, the header files above should be included in the origin and target program source files.

An additional header file (sil_defs.h) exists but should only be used within the build of the API libraries.

The header file calling order should be:

```
#include "sidefs.h"
#include "sitypes.h"
#include "sitypes2.h"
#include "dbvproto.h"
#include "iord_ext.h"
#include "itemordr.h"
```

Observation of the header files in the following sections will indicate #ifdef statements of the parameter PATCH. These apply to the API extensions which can be included as software extenxions to the API and should be available on the location where the API is retrieved from. Please refer to the software README files for installation of the main API software and on how to apply the software extensions in the patch form.

2 Definitions in sitypes.h

```
/* -----------  sitypes.h -------------------------------------------*/


/* When using this file with other DBV-OSI II includes, include as follows

#include "sidefs.h"
#include "sitypes.h"
#include "sitypes2.h"
#include "dbvproto.h"
#include "iord_ext.h"
#include "itemordr.h"
*/
#ifdef PATCH
/*
#include "crxtypes.h"
*/
#endif /* divert(0) */



/*  Module Name: @(#)sitypes.h      7.7
    Version: 4/24/97
    Author: Chris Attewell

    Crossnet Systems Ltd.
    Unit 3, The Paddock,
    Hambridge Road,
    Newbury,
    Berkshire, RG14 5TQ

    Description:
    Type definitions for DBV-OSI II

*/


#if !defined(SITYPES_H)
#define SITYPES_H

/* Responses for releasing an association */
typedef enum DbvReleaseRspReason {
      Normal = 0,
      Not_Finished = 1
} DbvReleaseRspReason;

typedef enum DbvReleaseRspResult {
      Affirmative = 0,
      Negative = 1
} DbvReleaseRspResult;

typedef struct DbvReleaseRsp {
      DbvReleaseRspReason      *peReleaseRspReason;
      DbvReleaseRspResult       eReleaseRspResult;
```

```
} DbvReleaseRsp;

/* Enumeration Types */

/* Events requiring a read operation:
typedef enum EventLocation
            API_No_Event (0),
            API_User_Event (1),
            API_Provider_Event (2)
} EventLocation;
typedef enum EventLocation {
            API_No_Event =0,
            API_User_Event =1,
            API_Provider_Event =2
} EventLocation;*/

/* New definition of DbvEventLocation */
typedef struct DbvEventLocation {
      DbvFds      **ppulUserEvent;
      DbvAssocId  **pplProviderEvent;
} DbvEventLocation;


/* Protocols supported: */
typedef enum DbvApplicationProtocol {
            Unknown_Protocol = 0,
            Unidentified_Protocol = 1,
            SR_Protocol = 2,
            Z3950_Protocol = 3
} DbvApplicationProtocol;

/* Protocol stacks supported: */
typedef enum DbvProtocolStack {
            Unknown_Stack = 0,
            OSI_Stack = 1,
            TCP_Stack = 2
} DbvProtocolStack;

/* Associate Response: */
typedef enum DbvAssociateRsp {
            Accepted = 0,
            Rejected_Perm = 1,
            Rejected_Tran = 2
} DbvAssociateRsp;

/* Abstract syntaxes: */
typedef enum DbvAbstractSyntax {
            No_Syntaxes_Provided = 0,
            No_Common_Syntax = 1,
            Common_Syntaxes = 2
} DbvAbstractSyntax;

/* Association states during association/connection establishment phase: */
typedef enum DbvAssociateState {
            Associate_Null  = 0,
            Associate_Accepted  = 1,
            Associate_Rejected  = 2,
```

```
            Associate_Pending  = 3
} DbvAssociateState;


/* Associate rejection: */
typedef enum DbvAssociateRejectSource {
            ACSE_User = 0,
            ACSE_Provider = 1
} DbvAssociateRejectSource;


typedef enum DbvAssociateRejectResult {
            Rejected_Permanent = 0,
            Rejected_Transient = 1
} DbvAssociateRejectResult;

typedef enum DbvAssociateRejectReason {
            No_Reason_Given = 0,
            No_Common_ACSE_Version = 1,
            Application_Context_Not_Supported = 2
} DbvAssociateRejectReason;


/* Association states during association/connection release phase: */
typedef enum DbvReleaseState {
            Release_Null  = 0,
            Release_Accepted  = 1,
            Release_Rejected  = 2,
            Release_Pending  = 3
} DbvReleaseState;

typedef enum DbvReleaseReq {
            ReleaseReq_Normal = 0,
            ReleaseReq_Urgent = 1,
            ReleaseReq_Undefined = 2
} DbvReleaseReq;


/* Association state upon return of an API function: */
typedef enum DbvAssociationState {
            Association_Open = 0,
            Association_Closed = 1,
            Association_Unknown_State = 2
} DbvAssociationState;


/* Error classes: */
typedef enum DbvErrorClass {
            SI_Communication_Error = 0,
            SI_Interface_Error = 1,
            SI_OperatingSystem_Error = 2,
            SI_Abort = 3
} DbvErrorClass;



/* Abort sources: */
typedef enum DbvAbortSource {
```

```
            SI_ACSE_User = 0,
            SI_ACSE_Provider = 1,
            SI_Presentation_User = 2,
            SI_Presentation_Provider = 3,
            SI_Transport_Provider = 4
} DbvAbortSource;

/* Permitted SR-/Z39.50 events at the Origin system */
typedef enum DbvOriginEvent {
            OR_InitializeConf = 0,
            OR_SearchConf = 1,
            OR_PresentConf = 2,
            OR_DeleteResultSetConf = 3,
            OR_ScanConf = 4,
            OR_ResourceReportConf = 5,
            OR_SegmentationInd = 6,
            OR_SortConf = 7,
            OR_ExtendedServicesConf = 8,
            OR_CloseConf = 9,
            OR_CloseInd = 10,
/*          OR_ReleaseInd = 11,  Target is not allowed to close the
                                     association */
            OR_Invalid_APDU = 11,
            OR_AccessCntrlInd = 12, /* AccessControlRequest from target */
            OR_ResourCntrlInd = 13  /* ResourceControlRequest from target */
} DbvOriginEvent;

/* Permitted SR-/Z39.50 events at the Target system */
typedef enum DbvTargetEvent {
            TA_InitializeInd = 0,
            TA_SearchInd = 1,
            TA_PresentInd = 2,
            TA_DeleteResultSetInd = 3,
            TA_ScanInd = 4,
            TA_ResourceReportInd = 5,
            TA_SortInd = 6,
            TA_ExtendedServicesInd = 7,
            TA_CloseInd = 8,
            TA_CloseConf = 9,
            TA_ReleaseInd = 10,
            TA_Invalid_APDU = 11,
            TA_AccessCntrlConf = 12, /* AccessControlResponse from origin */
            TA_ResourCntrlConf = 13,  /* ResourceControlResponse from origin */
            TA_TrigResourCntrlInd = 14  /* TriggerResourceControlRequest from
origin */

} DbvTargetEvent;


/* Target identification */
typedef enum DbvTargetNameOrAddress {
            Target_Name = 0,
            Target_Address = 1
} DbvTargetNameOrAddress;


/* Address types */
typedef enum DbvOsiOrInternetAddress {
```

```
            Osi_Address = 0,
            Internet_Address = 1
} DbvOsiOrInternetAddress;


typedef enum DbvTraceLevel {
      Trace_Off = 0,
      Trace_Low = 1,
      Trace_High = 2
} DbvTraceLevel;



/* Structure/Union Types  */

/* Structures associated with Association Handling functions */

typedef struct DbvAssociateReject {
      DbvAssociateRejectSource      *peAssociateRejectSource;
      DbvAssociateRejectResult      *peAssociateRejectResult;
      DbvAssociateRejectReason      *peAssociateRejectReason;
} DbvAssociateReject;

typedef struct DbvProtocolAndSyntaxes {
      DbvApplicationProtocol          *peApplicationProtocol;
      DbvAbstractSyntax               *peAbstractSyntax;
      char                          **ppcCommonSyntaxes;
/* valid <--> peAbstractSyntax == Common_Syntaxes */
} DbvProtocolAndSyntaxes;

typedef union DbvAssociateResult {
      DbvProtocolAndSyntaxes        *psProtocolAndSyntaxes;
      DbvAssociateReject            *psAssociateReject;
} DbvAssociateResult;


typedef struct DbvOsiAddress {
      char        *pcPsel;
      char        *pcSsel;
      char        *pcTsel;
      char        *pcNsap;
} DbvOsiAddress;

typedef struct DbvInternetAddress {
      char        *pcHostAddress;
      DbvPortNumber    ulPortNumber;
} DbvInternetAddress;

typedef struct DbvTargetInformation {
      DbvApplicationProtocol              *peTargetSupportedProtocol;
      DbvOsiOrInternetAddress               eOsiOrInternetAddress;
      union {
            DbvOsiAddress           *psOsiAddress;
            DbvInternetAddress      *psInternetAddress;
      }                               uOsiOrInternetAddress;
} DbvTargetInformation;
```

```
typedef struct DbvTarget {
      DbvTargetNameOrAddress        eTargetNameOrAddress;
      union {
            char                    *pcTargetName;
            DbvTargetInformation    *psTargetInformation;
      }                             uNameOrAddress;
} DbvTarget;
```

```
/* Structures common to all functions */

typedef struct DbvAbort {
  DbvAbortSource          eAbortSource;
  DbvAbortReason          lAbortReason;
  char                   *pcDiagnostics;
} DbvAbort;

typedef struct DbvError {
  DbvErrorClass         eErrorClass;
  union {
    DbvErrorCode         lIfOsCoError;
    DbvAbort            *psAbort;
  }                      uErrorCode;
  DbvAssociationState   eAssociationState;
} DbvError;
```

2.1 Types and Sub-Structures in sitypes.h

```
/* ------------------------------------------------------------*/
/*              Types and Sub-Structures                       */
/* ------------------------------------------------------------*/

typedef unsigned long   Ulong;
typedef char*           DbvProtocolVersion;
typedef char*           DbvOptions;


typedef enum DbvBool {
             DBV_FALSE = 0,
             DBV_TRUE  = 1
} DbvBool;


typedef char*           DbvObjectIdentifier;
```

2.1.1 DbvOctetString

```
/* -------------   OctetString   ----------------------------- */

typedef struct DbvOctetString {
Ulong      ulLength;
char      *pcValue;
} DbvOctetString;
```

2.1.2 DbvVisibleString

```
/* -------------   DbvVisibleString   ----------------------------- */

typedef char*          DbvVisibleString;
```

2.1.3 DbvGeneralizedTime

```
/* -------------   DbvGeneralizedTime   ----------------------------- */

typedef char*          DbvGeneralizedTime;
```

2.1.4 DbvInternationalString

```
/* -------------   DbvInternationalString   ----------------------------- */
typedef struct DbvInternationalString {
Ulong      ulLength;
char      *pcValue;
} DbvInternationalString;
```

2.1.5 DbvGeneralString

```
/* -------------   DbvGeneralString   ----------------------------- */

typedef struct DbvGeneralString {
Ulong      ulLength;
char      *pcValue;
} DbvGeneralString;
```

2.1.6 ASN1External

```
/* --------------  ASN1External  ----------------------- */

typedef enum DbvExternalValueType {
                     EVT_SingleASN1Type = 1,
                     EVT_OctetAligned   = 2,
                     EVT_Arbitrary      = 3
} DbvExternalValueType;

typedef enum DbvASN1Type {
             AT_UnstructuredText         = 1,
             AT_OpacRecord               = 2,
             AT_SummaryRecord            = 3,
             AT_ResourceReport1          = 4,
             AT_ResourceReport2          = 5,
             AT_EsTaskPackage            = 6,
             AT_UpdateExtendedService    = 7,
             AT_ItemOrderExtendedService = 9,
             AT_IllPdu                   = 10,
             AT_ExplainRecord            = 12,
#ifdef PATCH
             AT_GenericRecord            = 13,
#endif /* divert(0) */
             AT_PromptObject             = 14,
#ifdef PATCH
             AT_Print                = 15,
```

```
             AT_Alert                  = 16,
             AT_Corder                 = 17,
#endif /* divert(0) */
             AT_CharSetAndLanguage         = 18,
             AT_UnknownType                = 8
} DbvASN1Type;


typedef struct DbvSingleASN1Type {
DbvASN1Type          eASN1Type;
union {
    struct DbvInternationalString      *psUnstructuredText;
    struct DbvOpacRecord               *psOpacRecord;
    struct DbvSummaryRecord            *psSummaryRecord;
    struct DbvResourceReport1          *psResourceReport1;
    struct DbvResourceReport2          *psResourceReport2;
    struct DbvEsTaskPackage            *psEsTaskPackage;
    struct DbvUpdateExtendedService    *psUpdateExtendedService;
    struct DbvItemOrderExtendedService *psItemOrderExtendedService;
    struct DbvIllPdu                   *psIllPdu; /* ILL */
    struct DbvExplainRecord            *psExplainRecord;
#ifdef PATCH
    struct DbvGenericRecord            *psGenericRecord;
#endif /* divert(0) */
    struct DbvPromptObject             *psPromptObject;
#ifdef PATCH
    struct DbvPrintExtendedService     *psPrintExtendedService;
    struct DbvAlertExtendedService     *psAlertExtendedService;
    struct DbvCOrder                   *psCOrder;
#endif /* divert(0) */
    struct DbvCharSetAndLanguage       *psCharSetAndLanguage;
    struct DbvOctetString              *psUnknownType;
  }                uASN1;
} DbvSingleASN1Type;


typedef struct DbvASN1External {
DbvObjectIdentifier      pcObjectIdentifier;
DbvExternalValueType     eValueType;
union {
    DbvSingleASN1Type      *psSingleASN1Type;
    DbvOctetString         *psOctetAligned;
    DbvOctetString         *psArbitrary;
  }                    uExternalValue;
} DbvASN1External;
```

2.1.7 Other Information

```
/* -------------  Other Information  ---------------------- */

typedef enum DbvInformationType {
                   IT_CharacterInfo    = 1,
                   IT_BinaryInfo       = 2,
```

```
                    IT_ExternalInfo     = 3,
                    IT_ObjectIdentifier = 4
} DbvInformationType;


typedef struct DbvInfoCategory {
DbvObjectIdentifier    pcCategoryTypeId;
Ulong                  ulCategoryValue;
} DbvInfoCategory;


typedef struct DbvOtherInformation {
DbvInfoCategory       *psInfoCategory;
DbvInformationType     eInformationType;
union {
    DbvInternationalString      *psCharacterInfo;
    DbvOctetString              *psBinaryInfo;
    DbvASN1External             *psExternallyDefinedInfo;
    DbvObjectIdentifier          pcObjectIdentifier;
  }                   uInformation;
} DbvOtherInformation;
```

2.1.8 Authentication

```
/* --------------  Authentication ----------------------- */

typedef enum DbvAuthenticationType {
                 AT_Open       = 1,
                 AT_IdPassword = 2,
                 AT_Anonymous  = 3,
                 AT_Other      = 4
} DbvAuthenticationType;


typedef struct DbvIdPassword {
DbvInternationalString      *psGroupId;
DbvInternationalString      *psUserId;
DbvInternationalString      *psPassword;
} DbvIdPassword;


typedef struct DbvAuthentication{
DbvAuthenticationType   eAuthenticationType;
union {
    DbvVisibleString           psOpen;
    DbvIdPassword               *psIdPassword;
    DbvASN1External             *psOther;
  }                    uAuthentication;
} DbvAuthentication;
```

2.1.9 DbvElementSetNames

```
/* --------------  DbvElementSetNames  ----------------------- */

typedef enum DbvElementSetType {
                  EST_Generic          = 1,
                  EST_DatabaseSpecific = 2
} DbvElementSetType;


typedef struct DbvElementSetsDbSpecific {
DbvInternationalString       *psDatabaseName;
DbvInternationalString       *psElementSetName;
} DbvElementSetsDbSpecific;


typedef struct DbvElementSetNames {
DbvElementSetType      eElementSetType;
union {
    DbvInternationalString         *psGeneric;
    DbvElementSetsDbSpecific      **ppsElementSetsDbSpecific;
  }                uElementSet;
} DbvElementSetNames;
```

2.1.10 DbvQuery

```
/* --------------  DbvQuery  ----------------------- */


typedef enum DbvQueryType {
             QT_Private         = 1,
             QT_Rpn             = 2,
             QT_Iso8777         = 3,
             QT_Z3958           = 4,
             QT_Erpn            = 5,
             QT_NaturalLanguage = 6
} DbvQueryType;


typedef  enum DbvRPNStructureType {
                   RST_RpnLopRopOperator = 1,
                   RST_RpnOperand        = 2
} DbvRPNStructureType;


typedef enum DbvOperandType {
                  OT_AttributePlusTerm       = 1,
                  OT_ResultSetId             = 2,
                  OT_ResultSetPlusAttributes = 3
} DbvOperandType;
```

```
typedef enum DbvTermType {
             TT_General          = 1,    /* the original, Version 2 type */
             TT_Numeric          = 2,
             TT_SimpleCharacter  = 3,
             TT_ObjectIdentifier = 4,
             TT_DateTime         = 5,
             TT_External         = 6,
             TT_IntegerAndUnit   = 7,
             TT_Null             = 8
} DbvTermType;


typedef enum DbvOperatorType {
                  OT_Or        = 1,
                  OT_And       = 2,
                  OT_AndNot    = 3,
                  OT_Proximity = 4
} DbvOperatorType;

typedef enum DbvProximityUnitType {
                     PUT_Known   = 1,
                     PUT_Private = 2
} DbvProximityUnitType;


typedef enum DbvProximityRelationType {
                         PRT_LessThan             = 1,
                         PRT_LessThanOrEqual      = 2,
                         PRT_Equal                = 3,
                         PRT_GreaterThanOrEqual   = 4,
                         PRT_GreaterThan          = 5,
                         PRT_NotEqual             = 6,
                         PRT_Invalid              = -1
} DbvProximityRelationType;


typedef enum DbvKnownProximityUnit {
                      KPU_Character   = 1,
                      KPU_Word        = 2,
                      KPU_Sentence    = 3,
                      KPU_Paragraph   = 4,
                      KPU_Section     = 5,
                      KPU_Chapter     = 6,
                      KPU_Document    = 7,
                      KPU_Element     = 8,
                      KPU_SubElement  = 9,
                      KPU_ElementType = 10,
                      KPU_Byte        = 11,
                      KPU_Invalid     = -1
} DbvKnownProximityUnit;


typedef struct DbvProximityUnitCode {
DbvProximityUnitType    eProximityUnitType;
union {
    DbvKnownProximityUnit  eKnownProximityUnit;
    Ulong                  ulPrivate;
  }                  uProximityUnit;
```

```
} DbvProximityUnitCode;


typedef struct DbvProximityOperator {
DbvBool                   *pfExclusion;
Ulong                      ulDistance;
DbvBool                    fOrdered;
DbvProximityRelationType   eProximityRelationType;
DbvProximityUnitCode       sProximityUnitCode;
} DbvProximityOperator;


typedef struct DbvOperator {
DbvOperatorType       eOperatorType;
DbvProximityOperator *psProximityOperator; /* <--> eOperatorType == OT_Proximity
*/
} DbvOperator;


/* ---- DbvAttributeValue ----( not the final definition ) ---- */

typedef enum DbvAttributeValueType {
                 AVT_Numeric  = 1,    /* the original one */
                 AVT_Complex  = 2
} DbvAttributeValueType;
```

2.1.11 DbvIntUnit

```
/* -------------   DbvIntUnit   -------------------- */

typedef enum DbvStringOrNumericType {
                  SNT_String  = 1,
                  SNT_Numeric = 2
} DbvStringOrNumericType;


typedef struct DbvStringOrNumeric {
DbvStringOrNumericType     eStringOrNumeric;
union {
    DbvInternationalString    *psString;
    Ulong                      ulNumeric;
  }                      uStringOrNumeric;
} DbvStringOrNumeric;


typedef struct DbvUnit {
DbvInternationalString *psUnitSystem;
DbvStringOrNumeric     *psUnitType;
DbvStringOrNumeric     *psUnit;
Ulong                  *pulScaleFactor;
} DbvUnit;
```

```
typedef struct DbvIntUnit {
Ulong          ulValue;
DbvUnit        sUnitUsed;
} DbvIntUnit;


typedef struct DbvComplexValue {
DbvStringOrNumeric    **ppsList;
Ulong                 **ppulSemanticAction;
} DbvComplexValue;


typedef struct DbvAttributeValue {
DbvAttributeValueType          eAttributeValueType;
union {
    Ulong                     ulNumeric;
    DbvComplexValue          *psComplexValue;
  }                              uAttributeValue;
} DbvAttributeValue;


typedef struct DbvAttributeElement {
DbvObjectIdentifier      pcAttributeSetId;
Ulong                    ulAttributeType;
DbvAttributeValue        sAttributeValue;
} DbvAttributeElement;
```

2.1.12 DbvTerm

```
/* -------------   DbvTerm   ------------------ */

typedef struct DbvTerm {
DbvTermType        eTermType;
union {
    DbvOctetString              *psGeneral;
    Ulong                        ulInteger;
    DbvInternationalString      *psSimpleCharacter;
    DbvObjectIdentifier          pcObjectIdentifier;
    DbvGeneralizedTime           pcDateTime;
    DbvASN1External             *psExternal;
    DbvIntUnit                  *psIntegerAndUnit;
  }                uTerm;
} DbvTerm;


typedef struct DbvAttributesPlusTerm {
DbvAttributeElement    **ppsAttributeElement;
DbvTerm                  sTerm;
} DbvAttributesPlusTerm;


typedef struct DbvResultSetPlusAttributes {
```

```
DbvInternationalString       sResultSetId;
DbvAttributeElement        **ppsAttributeElement;
} DbvResultSetPlusAttributes;


typedef struct DbvOperand {
DbvOperandType   eOperandType;
union {
    DbvAttributesPlusTerm      *psAttributesPlusTerm;
    DbvInternationalString     *psResultSetId;
    DbvResultSetPlusAttributes *psResultSetPlusAttributes;
  }            uOperand;
} DbvOperand;


typedef struct DbvRPNLopRopOperator {
struct DbvRPNStructure  *psLeftOperand;
struct DbvRPNStructure  *psRightOperand;
DbvOperator              sOperator;
} DbvRPNLopRopOperator;


typedef struct DbvRPNStructure {
DbvRPNStructureType   eRpnStructureType;
union {
    DbvRPNLopRopOperator       *psRpnLopRopOperator;
    DbvOperand                 *psRpnOperand;
  }                uRPNStructure;
} DbvRPNStructure;



typedef struct DbvRPNQuery {
DbvObjectIdentifier         pcAttributeSetId;
DbvRPNStructure             sRpnStructure;
} DbvRPNQuery;
```

2.1.13 DbvQuery

```
/* --------------  DbvQuery  ----------------------- */

typedef struct DbvQuery {
DbvQueryType        eQueryType;
union {
    DbvOctetString          *psPrivateQuery;
    DbvRPNQuery             *psRpnQuery;
    DbvOctetString          *psIso8777Query;
    DbvOctetString          *psZ3958Query;
    DbvRPNQuery             *psErpnQuery;
    DbvOctetString          *psNaturalLanguageQuery;
  }              uQuery;
} DbvQuery;
```

2.1.14 DbvRecords

```
/* --------------  DbvRecords  ----------------------- */

typedef enum DbvRecordType {
              RT_Response                  = 1,
              RT_NonSurrDiagnostics        = 2,
              RT_MultipleNonSurrDiagnostics = 3
} DbvRecordType;


typedef enum DbvDbRecordType {
                DRT_Retrieval            = 1,
                DRT_SurrogateDiagnostics = 2,
                DRT_StartingFragment     = 3,
                DRT_IntermediateFragment = 4,
                DRT_FinalFragment        = 5
} DbvDbRecordType;


typedef enum DbvDiagnosticsType {
                   DT_DefaultFormat     = 1,
                   DT_ExternallyDefined = 2
} DbvDiagnosticsType;


typedef enum DbvFragmentSyntaxType {
                      FST_ExternallyTagged    = 1,
                      FST_NotExternallyTagged = 2
} DbvFragmentSyntaxType;


typedef enum DbvProtocolVersionInForce {
                      PVIF_Version2 = 1,
                      PVIF_Version3 = 2
} DbvProtocolVersionInForce;


typedef struct DbvAdditionalInfo {
DbvProtocolVersionInForce   eProtocolVersionInForce;
union {
    DbvVisibleString            pcV2;
    DbvInternationalString     *psV3;
  }                          uAdditionalInfo;
} DbvAdditionalInfo;

typedef struct DbvDefaultDiagFormat {
```

```
DbvObjectIdentifier        pcDiagnosticSetId;
Ulong                      ulCondition;
DbvAdditionalInfo          sAdditionalInfo;
} DbvDefaultDiagFormat;


typedef struct DbvDiagRec {
DbvDiagnosticsType       eDiagnosticsType;
union {
    DbvDefaultDiagFormat      *psDefault;
    DbvASN1External           *psExternallyDefined;
  }                   uDiagnostics;
} DbvDiagRec;


typedef struct DbvFragmentSyntax {
DbvFragmentSyntaxType      eFragmentSyntaxType;
union {
    DbvASN1External    *psExternallyTagged;
    DbvOctetString     *psNotExternallyTagged;
  }                     uFragmentSyntax;
} DbvFragmentSyntax;


typedef struct DbvNamePlusRecord {
DbvInternationalString  *psDatabaseName;
DbvDbRecordType           eDbRecordType;
union {
    DbvASN1External   *psRetrievalRecord;
    DbvDiagRec        *psSurrogateDiagnostic;
    DbvFragmentSyntax *psStartingFragment;
    DbvFragmentSyntax *psIntermediateFragment;
    DbvFragmentSyntax *psFinalFragment;
  }                   uDbRecord;
} DbvNamePlusRecord;



typedef struct DbvRecords {
DbvRecordType        eRecordType;
union {
    DbvNamePlusRecord    **ppsResponseRecord;
    DbvDefaultDiagFormat  *psNonSurrogateDiag;
    DbvDiagRec           **ppsMultipleNonSurrogateDiag;
  }               uRecord;
} DbvRecords;
```

2.1.15 DbvRange

```
/* --------------  DbvRange  ----------------------- */

typedef struct DbvRange {
```

```
Ulong        ulStartingPosition;
Ulong        ulNumberOfRecords;
} DbvRange;
```

2.1.16 Record Compsition

```
/* --------------  RecordComposition  ----------------------- */

typedef enum DbvRecordCompositionType {
                          RCT_ElementSetNames = 1,
                          RCT_CompSpec        = 2
} DbvRecordCompositionType;

typedef enum DbvElementSpecType {
                   EST_Primitive = 1,
                   EST_External  = 2
} DbvElementSpecType;

typedef struct DbvElementSpec {
DbvElementSpecType        eElementSpecType;
union {
    DbvInternationalString  *psPrimitive;
    DbvASN1External         *psExternal;
  }                    uElementSpec;
} DbvElementSpec;

typedef struct DbvSpecification {
DbvObjectIdentifier        pcSchema;
DbvElementSpec            *psElementSpec;
} DbvSpecification;

typedef struct DbvDbSpecificComp {
DbvInternationalString    sDatabaseName;
DbvSpecification          sSpecification;
} DbvDbSpecificComp;

typedef struct DbvCompSpec {
DbvBool                  fSelectAlternativeSyntax;
DbvSpecification        *psGeneric;
DbvDbSpecificComp      **ppsDbSpecific;
DbvObjectIdentifier     *ppcRecordSyntax;
} DbvCompSpec;
```

```
typedef struct DbvRecordComposition {
DbvRecordCompositionType    eRecordCompositionType;
union {
    DbvElementSetNames  *psElementSetNames;
    DbvCompSpec         *psCompSpec;
  }                      uRecordComposition;
} DbvRecordComposition;
```

2.1.17 List Status

```
/* ----------  ListStatus  (--> DELETE-RESULT-SET response) --------- */

typedef enum DbvDeleteSetStatus {
                   DSS_Success                     = 0,
                   DSS_ResultSetDidNotExist        = 1,
                   DSS_PreviouslyDeletedByTarget   = 2,
                   DSS_SystemProblemAtTarget       = 3,
                   DSS_AccessNotAllowed            = 4,
                   DSS_ResourceControlAtOrigin     = 5,
                   DSS_ResourceControlAtTarget     = 6,
                   DSS_BulkDeleteNotSupported      = 7,
                   DSS_NotAllSetsDeletedOnBulkDlt  = 8,
                   DSS_NotAllRequestedSetsDeleted  = 9,
                   DSS_ResultSetInUse              = 10,
                   DSS_Invalid                     = -1
} DbvDeleteSetStatus;


typedef struct DbvListStatus {
DbvInternationalString       sResultSetId;
DbvDeleteSetStatus           eDeleteSetStatus;
} DbvListStatus;
```

2.1.18 DbvListEntries (SCAN)

```
/* --------------  DbvListEntries  (--> SCAN ) ---------------- */


typedef enum DbvEntryType {
             ET_TermInfo            = 1,
             ET_SurrogateDiagnostic = 2
} DbvEntryType;


typedef enum DbvOccurrenceType {
```

```
                  OT_Global     = 1,
                  OT_ByDatabase = 2
} DbvOccurrenceType;


typedef struct DbvOccurrenceByDb {
DbvInternationalString       sDatabaseName;
Ulong                       *pulNumber;
DbvOtherInformation        **ppsOtherInformation;
} DbvOccurrenceByDb;


typedef struct DbvOccurrence {
DbvOccurrenceType             eOccurrenceType;
union {
    Ulong                ulGlobal;
    DbvOccurrenceByDb  **ppsOccurrenceByDb;
  }                           uOccurence;
} DbvOccurrence;


typedef struct DbvOccurrenceByAttributes {
DbvAttributeElement      **ppsAttributeElement;
DbvOccurrence             *psOccurrence;
DbvOtherInformation      **ppsOtherInformation;
} DbvOccurrenceByAttributes;


typedef struct DbvTermInfo {
DbvTerm                      sTerm;
DbvInternationalString      *psDisplayTerm;
DbvAttributeElement        **ppsSuggestedAttribute;
DbvAttributesPlusTerm      **ppsAlternativeTerm;
Ulong                       *pulGlobalOccurrences;
DbvOccurrenceByAttributes **ppsByAttributes;
DbvOtherInformation        **ppsOtherInformation;
} DbvTermInfo;


typedef struct DbvEntry {
DbvEntryType         eEntryType;
union {
    DbvTermInfo    *psTermInfo;
    DbvDiagRec     *psSurrogateDiagnostic;
  }               uEntry;
} DbvEntry;


typedef struct DbvListEntries {
DbvEntry    **ppsEntry;
DbvDiagRec  **ppsNonSurrogateDiagnostics;
} DbvListEntries;
```

2.1.19 Sort Sequence

```
/* --------------  SortSequence  ----------------------- */

typedef enum DbvSortElementType {
                     SET_Generic    = 1,
                     SET_DbSpecific = 2
} DbvSortElementType;


typedef enum DbvSortRelationType {
                    SRT_Ascending     = 1,
                    SRT_Descending    = 2,
                    SRT_FrequencyAsc  = 3,
                    SRT_FrequencyDesc = 4,
                    SRT_Invalid       = -1
} DbvSortRelationType;


typedef enum DbvSortCaseType {
                SCT_CaseSensitive   = 1,
                SCT_CaseInsensitive = 2,
                SCT_Invalid         = -1
} DbvSortCaseType;


typedef enum DbvSortKeyType {
                    SKT_SortField      = 1,
                    SKT_ElementSpec    = 2,
                    SKT_SortAttributes = 3
} DbvSortKeyType;


typedef enum DbvSortMVActionType {
                    SMT_Abort    = 1,
                    SMT_Null     = 2,
                    SMT_Constant = 3
} DbvSortMVActionType;


typedef struct DbvSortAttributes {
DbvObjectIdentifier     pcAttributeSetId;
DbvAttributeElement   **ppsAttributeElement;
} DbvSortAttributes;


typedef struct DbvSortKey {
DbvSortKeyType              eSortKeyType;
union {
    DbvInternationalString     *psSortField;
    DbvSpecification           *psElementSpec;
    DbvSortAttributes          *psSortAttributes;
  }                        uSortElement;
} DbvSortKey;


typedef struct DbvSortMVAction {
```

```
DbvSortMVActionType    eSortMVActionType;
DbvOctetString        *psOctetString;  /* <--> eSortMVActionType == SMT_Constant
*/
} DbvSortMVAction;


typedef struct DbvDbSpecificSortKey {
DbvInternationalString      sDatabaseName;
DbvSortKey                  sDbSort;
} DbvDbSpecificSortKey;


typedef struct DbvSortElement {
DbvSortElementType        eSortElementType;
union {
    DbvSortKey                *psGeneric;
    DbvDbSpecificSortKey     **ppsDatabaseSpecific;
  }                       uSortElement;
} DbvSortElement;


typedef struct DbvSortKeySpec {
DbvSortElement         sSortElement;
DbvSortRelationType    eSortRelationType;
DbvSortCaseType        eSortCaseType;
DbvSortMVAction       *psMVActionType;
} DbvSortKeySpec;
```

2.1.20 Origin Supplied Common Parameters - Extended Services

```
/* ---  OriginSuppliedCommonParameters  (--> EXTENDED-SERVICES) ----------- */

typedef enum DbvPermAllowableFunctions {
                      PAF_Delete            = 1,
                      PAF_ModifyContents    = 2,
                      PAF_ModifyPermissions = 3,
                      PAF_Present           = 4,
                      PAF_Invoke            = 5,
                      PAF_Invalid           = -1
} DbvPermAllowableFunctions;


typedef struct DbvPermissions {
DbvInternationalString          sUserId;
DbvPermAllowableFunctions     **ppeAllowableFunctions;
} DbvPermissions;
```

2.2 PDU Data Structures

```
/* -------------------------------------------------------------------*/
/*                 PDU-Datastructures      (Top-Level structures)    */
/* -------------------------------------------------------------------*/
```

2.2.1 Initialize

```
/* ------------------    INITIALIZE   --------------------------------- */


typedef struct DbvInitializeReq {
DbvOctetString         *psReferenceId;
DbvProtocolVersion      pcProtocolVersion;
DbvOptions              pcOptions;
Ulong                   ulPreferredMessageSize;
Ulong                   ulMaximumMessageSize;
DbvAuthentication      *psAuthentication;
DbvInternationalString *psImplementationId;
DbvInternationalString *psImplementationName;
DbvInternationalString *psImplementationVersion;
DbvASN1External        *psUserInformationField;
DbvOtherInformation   **ppsOtherInformation;
} DbvInitializeReq;



typedef struct DbvInitializeRsp {
DbvOctetString         *psReferenceId;
DbvProtocolVersion      pcProtocolVersion;
DbvOptions              pcOptions;
Ulong                   ulPreferredMessageSize;
Ulong                   ulMaximumMessageSize;
DbvBool                 fResult;
DbvInternationalString *psImplementationId;
DbvInternationalString *psImplementationName;
DbvInternationalString *psImplementationVersion;
DbvASN1External        *psUserInformationField;
DbvOtherInformation   **ppsOtherInformation;
} DbvInitializeRsp;
```

2.2.2 Search

```
/* ------------------    SEARCH   --------------------------------- */

typedef enum DbvResultSetStatus {
                   RSS_Subset  = 1,
```

```
                   RSS_Interim = 2,
                   RSS_None    = 3,
                   RSS_Invalid = -1
} DbvResultSetStatus;


typedef enum DbvPresentStatus {
                 PS_Success   = 0,
                 PS_Partial_1 = 1,
                 PS_Partial_2 = 2,
                 PS_Partial_3 = 3,
                 PS_Partial_4 = 4,
                 PS_Failure   = 5,
                 PS_Invalid   = -1
} DbvPresentStatus;



typedef struct DbvSearchReq {
DbvOctetString            *psReferenceId;
Ulong                      ulSmallSetUpperBound;
Ulong                      ulLargeSetLowerBound;
Ulong                      ulMediumSetPresentNumber;
DbvBool                    fReplaceIndicator;
DbvInternationalString     sResultSetName;
DbvInternationalString   **ppsDatabaseNames;
DbvElementSetNames        *psSmallSetElementNames;
DbvElementSetNames        *psMediumSetElementNames;
DbvObjectIdentifier        pcPreferredRecordSyntax;
DbvQuery                  *psQuery;
DbvOtherInformation      **ppsAdditionalSearchInfo;
DbvOtherInformation      **ppsOtherInformation;
} DbvSearchReq;


typedef struct DbvSearchRsp {
DbvOctetString          *psReferenceId;
Ulong                    ulResultCount;
Ulong                    ulNumberOfRecordsReturned;
Ulong                    ulNextResultSetPosition;
DbvBool                  fSearchStatus;
DbvResultSetStatus      *peResultSetStatus;
DbvPresentStatus        *pePresentStatus;
DbvRecords              *psRecords;
DbvOtherInformation    **ppsAdditionalSearchInfo;
DbvOtherInformation    **ppsOtherInformation;
} DbvSearchRsp;
```

2.2.3 Present

```
/* ------------------    PRESENT   --------------------------------- */

typedef struct DbvPresentReq{
DbvOctetString              *psReferenceId;
```

```
DbvInternationalString      sResultSetId;
Ulong                       ulResultSetStartPoint;
Ulong                       ulNumberOfRecordsRequested;
DbvRange                  **ppsAdditionalRanges;
DbvRecordComposition       *psRecordComposition;
DbvObjectIdentifier         pcPreferredRecordSyntax;
Ulong                      *pulMaxSegmentCount;
Ulong                      *pulMaxRecordSize;
Ulong                      *pulMaxSegmentSize;
DbvOtherInformation       **ppsOtherInformation;
} DbvPresentReq;


typedef struct DbvPresentRsp {
DbvOctetString         *psReferenceId;
Ulong                   ulNumberOfRecordsReturned;
Ulong                   ulNextResultSetPosition;
DbvPresentStatus        ePresentStatus;
DbvRecords             *psRecords;
DbvOtherInformation   **ppsOtherInformation;
} DbvPresentRsp;
```

2.2.4 Delete Result Set

```
/* ------------------    DELETE-RESULT-SET   ----------------------------- */


typedef enum DbvDeleteFunction {
                  DF_List    = 0,
                  DF_All     = 1,
                  DF_Invalid = -1
} DbvDeleteFunction;


typedef struct DbvDeleteResultSetReq {
DbvOctetString               *psReferenceId;
DbvDeleteFunction             eDeleteFunction;
DbvInternationalString      **ppsResultSetId;
DbvOtherInformation         **ppsOtherInformation;
} DbvDeleteResultSetReq;


typedef struct DbvDeleteResultSetRsp {
DbvOctetString             *psReferenceId;
DbvDeleteSetStatus          eDeleteOperationStatus;
DbvListStatus             **ppsDeleteListStatuses;
Ulong                      *pulNumberNotDeleted;
DbvListStatus             **ppsBulkStatuses;
DbvInternationalString     *psDeleteMessage;
DbvOtherInformation       **ppsOtherInformation;
} DbvDeleteResultSetRsp;
```

2.2.5 Segment

```
/* ------------------    SEGMENT   ----------------------------- */

typedef struct DbvSegmentReq {
DbvOctetString         *psReferenceId;
Ulong                   ulNumberOfRecordsReturned;
DbvNamePlusRecord     **ppsSegmentRecords;
DbvOtherInformation   **ppsOtherInformation;
} DbvSegmentReq;
```

2.2.6 Resource Report

```
/* ------------------    RESOURCE-REPORT   ----------------------------- */

typedef enum DbvResourceReportStatus {
                        RRS_Success   = 0,
                        RRS_Partial   = 1,
                        RRS_Failure_1 = 2,
                        RRS_Failure_2 = 3,
                        RRS_Failure_3 = 4,
                        RRS_Failure_4 = 5,
                        RRS_Failure_5 = 6,
                        RRS_Failure_6 = 7,
                        RRS_Invalid   = -1
} DbvResourceReportStatus;


typedef struct DbvResourceReportReq {
DbvOctetString            *psReferenceId;
DbvOctetString            *psOperationId;
DbvObjectIdentifier        pcPrefResourceReportFormat;
DbvOtherInformation      **ppsOtherInformation;
} DbvResourceReportReq;


typedef struct DbvResourceReportRsp {
DbvOctetString             *psReferenceId;
DbvResourceReportStatus     eResourceReportStatus;
DbvASN1External            *psResourceReport;
```

```
DbvOtherInformation       **ppsOtherInformation;
} DbvResourceReportRsp;
```

2.2.7 Scan

```
/* ------------------    SCAN   ----------------------------- */

typedef enum DbvScanStatus {
              SCS_Success   = 0,
              SCS_Partial_1 = 1,
              SCS_Partial_2 = 2,
              SCS_Partial_3 = 3,
              SCS_Partial_4 = 4,
              SCS_Partial_5 = 5,
              SCS_Failure   = 6,
              SCS_Invalid   = -1
} DbvScanStatus;


typedef struct DbvScanReq {
DbvOctetString               *psReferenceId;
DbvInternationalString      **ppsDatabaseNames;
DbvObjectIdentifier           pcAttributeSetId;
DbvAttributesPlusTerm         sAttributesPlusTerm;
Ulong                        *pulStepSize;
Ulong                         ulNumberOfTermsRequested;
Ulong                        *pulPreferredPositionInResponse;
DbvOtherInformation         **ppsOtherInformation;
} DbvScanReq;


typedef struct DbvScanRsp {
DbvOctetString          *psReferenceId;
Ulong                   *pulStepSize;
DbvScanStatus            eScanStatus;
Ulong                    ulNumberOfEntriesReturned;
Ulong                   *pulPositionOfTerm;
DbvListEntries          *psEntries;
DbvObjectIdentifier      pcAttributeSetId;
DbvOtherInformation    **ppsOtherInformation;
} DbvScanRsp;
```

2.2.8 Sort

```
/* ------------------    SORT   ----------------------------- */


typedef enum DbvSortStatus {
                       SOS_Success   = 0,
                       SOS_Partial_1 = 1,
                       SOS_Failure   = 2,
                       SOS_Invalid   = -1
} DbvSortStatus;


typedef enum DbvSortResultSetStatus {
                       SRSS_Empty     = 0,
                       SRSS_Interim   = 1,
                       SRSS_Unchanged = 2,
                       SRSS_None      = 3,
                       SRSS_Invalid   = -1
} DbvSortResultSetStatus;



typedef struct DbvSortReq {
DbvOctetString           *psReferenceId;
DbvInternationalString  **ppsInputResultSetName;
DbvInternationalString    sSortedResultSetName;
DbvSortKeySpec          **ppsSortSequence;
DbvOtherInformation     **ppsOtherInformation;
} DbvSortReq;


typedef struct DbvSortRsp {
DbvOctetString             *psReferenceId;
DbvSortStatus               eSortStatus;
DbvSortResultSetStatus     *peResultSetStatus;
DbvDiagRec                **ppsDiagnostics;
DbvOtherInformation       **ppsOtherInformation;
} DbvSortRsp;
```

2.2.9 Close

```
/* ------------------    CLOSE   ----------------------------- */


typedef enum DbvCloseReason {
               CR_Finished          = 0,
               CR_ShutDown          = 1,
               CR_SystemProblem     = 2,
               CR_CostLimit         = 3,
               CR_Resources         = 4,
```

```
                CR_SecurityViolation = 5,
                CR_ProtocolError     = 6,
                CR_LackOfActivity    = 7,
                CR_PeerAbort         = 8,
                CR_Unspecified       = 9,
                CR_Invalid           = -1
} DbvCloseReason;



typedef struct DbvCloseReq {
DbvOctetString                    *psReferenceId;
DbvCloseReason                     eCloseReason;
DbvInternationalString            *psDiagnosticInformation;
DbvObjectIdentifier                pcResourceReportFormat;
DbvASN1External                   *psResourceReport;
DbvOtherInformation              **ppsOtherInformation;
} DbvCloseReq;
```

2.2.10 Extended Services

```
/* ------------------    EXTENDED-SERVICES   ----------------------------- */

typedef enum DbvExtServicesFunctions {
                   ESF_Create = 1,
                   ESF_Delete = 2,
                   ESF_Modify = 3,
                   ESF_Invalid     = -1
} DbvExtServicesFunctions;


typedef enum DbvExtServicesWaitAction {
                   ESWA_Wait                = 1,
                   ESWA_WaitIfPossible      = 2,
                   ESWA_DontWait            = 3,
                   ESWA_DontReturnPackage   = 4,
                   ESWA_Invalid             = -1
} DbvExtServicesWaitAction;


typedef enum DbvExtServicesOperationStatus {
                   ESOS_Done     = 1,
                   ESOS_Queued   = 2,
                   ESOS_Failure  = 3,
                   ESOS_Invalid  = -1
} DbvExtServicesOperationStatus;


typedef struct DbvExtendedServicesReq {
DbvOctetString                       *psReferenceId;
```

```
DbvExtServicesFunctions              eFunction;
DbvObjectIdentifier                  pcPackageType;
DbvInternationalString              *psPackageName;
DbvInternationalString              *psUserId;
DbvIntUnit                          *psRetentionTime;
DbvPermissions                     **ppsPermissions;
DbvInternationalString              *psDescription;
DbvASN1External                     *psOriginSpecificParameter;
DbvExtServicesWaitAction             eWaitAction;
DbvInternationalString              *psElements;
DbvOtherInformation                **ppsOtherInformation;
} DbvExtendedServicesReq;


typedef struct DbvExtendedServicesRsp {
DbvOctetString                     *psReferenceId;
DbvExtServicesOperationStatus       eOperationStatus;
DbvDiagRec                        **ppsDiagnostics;
DbvASN1External                    *psTaskPackage;
DbvOtherInformation               **ppsOtherInformation;
} DbvExtendedServicesRsp;
```

2.2.11 OID, Record Syntaxes and Resource Report Formats

```
/* ------------------------------------------------------------*/
/*   Object Identifier, Record Syntaxes, Resource Report      */
/* ------------------------------------------------------------*/


/* ---------------------------------------------------------------*/
/* The syntax Object Identifier definitions is

   ObjectIdentifier ::= Header Trailer
   Header ::= "1.0.10163." | "1.2.840.10003."
   Trailer ::= Number"."Number"}"
   Number  ::= positive integer value


 ---------------------------------------------------------------*/


/* -------------- Object Identifier (Record Syntaxes) -----------*/


#define ISO_UNIMARC                   "1.0.10163.5.1"
#define ANSI_UNIMARC                  "1.2.840.10003.5.1"
#define ISO_INTERMARC                 "1.0.10163.5.2"
#define ANSI_INTERMARC                "1.2.840.10003.5.2"
```

```
#define ISO_CCF                          "1.0.10163.5.3"
#define ANSI_CCF                         "1.2.840.10003.5.3"
#define ISO_USMARC                       "1.0.10163.5.10"
#define ANSI_USMARC                      "1.2.840.10003.5.10"
#define ISO_UKMARC                       "1.0.10163.5.11"
#define ANSI_UKMARC                      "1.2.840.10003.5.11"
#define ISO_NORMARC                      "1.0.10163.5.12"
#define ANSI_NORMARC                     "1.2.840.10003.5.12"
#define ISO_LIBRISMARC                   "1.0.10163.5.13"
#define ANSI_LIBRISMARC                  "1.2.840.10003.5.13"
#define ISO_DANMARC                      "1.0.10163.5.14"
#define ANSI_DANMARC                     "1.2.840.10003.5.14"
#define ISO_FINMARC                      "1.0.10163.5.15"
#define ANSI_FINMARC                     "1.2.840.10003.5.15"
#define ISO_MAB                          "1.0.10163.5.16"
#define ANSI_MAB                               "1.2.840.10003.5.16"
#define ISO_CANMARC                            "1.0.10163.5.17"
#define ANSI_CANMARC                           "1.2.840.10003.5.17"
#define ISO_SBN                                "1.0.10163.5.18"
#define ANSI_SBN                               "1.2.840.10003.5.18"
#define ISO_PICAMARC                           "1.0.10163.5.19"
#define ANSI_PICAMARC                          "1.2.840.10003.5.19"
#define ISO_AUSMARC                            "1.0.10163.5.20"
#define ANSI_AUSMARC                           "1.2.840.10003.5.20"

#define ISO_EXPLAIN_RECORD                     "1.0.10163.5.100"
#define ANSI_EXPLAIN_RECORD                    "1.2.840.10003.5.100"
#define ISO_UNSTRUCTURED_TEXT_RECORD           "1.0.10163.5.101"
#define ANSI_UNSTRUCTURED_TEXT_RECORD          "1.2.840.10003.5.101"
#define ISO_OPAC_RECORD                        "1.0.10163.5.102"
#define ANSI_OPAC_RECORD                       "1.2.840.10003.5.102"
#define ISO_SUMMARY_RECORD                     "1.0.10163.5.103"
#define ANSI_SUMMARY_RECORD                    "1.2.840.10003.5.103"
#define ISO_GENERIC_RECORD                     "1.0.10163.5.105"
#define ANSI_GENERIC_RECORD                    "1.2.840.10003.5.105"
#define ISO_ES_TASK_PACKAGE_RECORD             "1.0.10163.5.106"
#define ANSI_ES_TASK_PACKAGE_RECORD            "1.2.840.10003.5.106"



/* -------------- Object Identifier (Attribute-Set) -----------*/

#define ISO_ATTRIBUTE_SET_BIB_1                "1.0.10163.3.1"
#define ANSI_ATTRIBUTE_SET_BIB_1               "1.2.840.10003.3.1"


#define ANSI_ATTRIBUTE_SET_EXP_1               "1.2.840.10003.3.2"
#define ANSI_ATTRIBUTE_SET_CIP_1               "1.2.840.10003.3.2000.99.1"



/* -------------- Object Identifier (Diagnostic-Set) -----------*/

#define ISO_DIAGNOSTIC_SET_BIB_1               "1.0.10163.4.1"
#define ANSI_DIAGNOSTIC_SET_BIB_1              "1.2.840.10003.4.1"
```

```
/* -------------- Object Identifier (Resource-Report) -----------*/

#define ISO_RESOURCE_REPORT_1            "1.0.10163.7.1"
#define ANSI_RESOURCE_REPORT_1           "1.2.840.10003.7.1"
#define ISO_RESOURCE_REPORT_2            "1.0.10163.7.2"
#define ANSI_RESOURCE_REPORT_2           "1.2.840.10003.7.2"


#ifdef PATCH
/* -------------- Object Identifier (PERSISTENT QUERY Extended Service) -----*/
#define ISO_PERSISTENTQUERY_EXTENDED_SERVICE    "1.0.10163.9.2"
#define ANSI_PERSISTENTQUERY_EXTENDED_SERVICE   "1.2.840.10003.9.2"

#endif /* divert(0) */

/* -------------- Object Identifier (ITEM ORDER Extended Service) -----------*/
#define ISO_ITEMORDER_EXTENDED_SERVICE          "1.0.10163.9.4"
#define ANSI_ITEMORDER_EXTENDED_SERVICE         "1.2.840.10003.9.4"


/* -------------- Object Identifier (ILL PDU) -----------*/

#define ISO_ILL_PDU                             "1.2.10161.2.1"


/* ILL DbvExtension ulIdentifier int->ANY mappings */

#define SUBITO_ADDITIONAL_INFO                  800


/* -------------- Object Identifier (UPDATE Extended Service) -----------*/

#define ISO_UPDATE_EXTENDED_SERVICE             "1.0.10163.9.5"
#define ANSI_UPDATE_EXTENDED_SERVICE            "1.2.840.10003.9.5"


/* -------------- Object Identifier (Access Control Formats) -----------*/

#define ISO_ACCESS_PROMPT_1                     "1.0.10163.8.1"
#define ANSI_ACCESS_PROMPT_1                    "1.2.840.10003.8.1"



#ifdef PATCH

/* ---- Object Identifier (Crossnet developed for ESA (Esrin) ) ---- */

/* 70 = implementor id of ESA (Esrin) */
```

```
#define ESRIN_PRINT                    "1.2.840.10003.9.1000.70.1"
#define ESRIN_ALERT                    "1.2.840.10003.9.1000.70.2"
#define ESRIN_ORDER                    "1.2.840.10003.9.4.1000.70.1"

#endif /* divert(0) */

/* -------------- Object Identifier (User Information Formats) -----------*/

#define ISO_CHARSET_AND_LANGUAGE_NEGOTIATION         "1.0.10163.10.2"
#define ANSI_CHARSET_AND_LANGUAGE_NEGOTIATION        "1.2.840.10003.10.2"



/* -------------- Record Syntax "UnstructuredText" ----------------*/

typedef DbvInternationalString   DbvUnstructuredText;
```

2.2.12 OPAC Record

```
/* -------------- Record Syntax "OpacRecord" ----------------*/

typedef struct DbvCircRecord {
DbvBool                      fAvailableNow;
DbvInternationalString     *psAvailabilityDate;
DbvInternationalString     *psAvailableThru;
DbvInternationalString     *psRestrictions;
DbvInternationalString     *psItemId;
DbvBool                      fRenewable;
DbvBool                      fOnHold;
DbvInternationalString     *psEnumAndChron;
DbvInternationalString     *psMidspine;
DbvInternationalString     *psTemporaryLocation;
} DbvCircRecord;


typedef struct DbvVolume {
DbvInternationalString        *psEnumeration;
DbvInternationalString        *psChronology;
DbvInternationalString        *psEnumAndChron;
} DbvVolume;


typedef struct DbvHoldingsAndCircData {
DbvInternationalString        *psTypeOfRecord;
DbvInternationalString        *psEncodingLevel;
DbvInternationalString        *psFormat;
DbvInternationalString        *psReceiptAcqStatus;
DbvInternationalString        *psGeneralRetention;
DbvInternationalString        *psCompleteness;
DbvInternationalString        *psDateOfReport;
DbvInternationalString        *psNucCode;
DbvInternationalString        *psLocalLocation;
```

```
DbvInternationalString        *psShelvingLocation;
DbvInternationalString        *psCallNumber;
DbvInternationalString        *psShelvingData;
DbvInternationalString        *psCopyNumber;
DbvInternationalString        *psPublicNote;
DbvInternationalString        *psReproductionNote;
DbvInternationalString        *psTermsUseRepro;
DbvInternationalString        *psEnumAndChron;
DbvVolume                    **ppsVolume;
DbvCircRecord                **ppsCircRecord;
} DbvHoldingsAndCircData;



typedef enum DbvHoldingsRecType {
      HRT_MarcHoldingsRecord  = 1,
      HRT_HoldingsAndCirc     = 2
} DbvHoldingsRecType;


typedef struct DbvHoldingsRecord {
DbvHoldingsRecType      eHoldingsRecType;
union {
    DbvASN1External          sMarcHoldingsRecord;
    DbvHoldingsAndCircData  sHoldingsAndCirc;
  }                     uMarcOrHoldAndCirc;
} DbvHoldingsRecord;

typedef struct DbvOpacRecord {
DbvASN1External         *psBibliographicRecord;
DbvHoldingsRecord      **ppsHoldingsData;
} DbvOpacRecord;
```

2.2.13 Summary Record

```
/* -------------- Record Syntax "SummaryRecord" ----------------*/

typedef struct DbvFormatSpec {
DbvInternationalString          sType;
Ulong                          *pulSize;
Ulong                          *pulBestPosn;
} DbvFormatSpec;


typedef struct DbvSummaryRecord {
DbvInternationalString           sTitle;
DbvInternationalString          *psAuthor;
DbvInternationalString          *psCallNumber;
DbvInternationalString          *psRecordType;
DbvInternationalString          *psBibliographicLevel;
DbvFormatSpec                  **ppsFormat;
DbvInternationalString          *psPublicationPlace;
DbvInternationalString          *psPublicationDate;
```

```
DbvInternationalString          *psTargetSystemKey;
DbvInternationalString          *psSatisfyingElement;
Ulong                           *pulRank;
DbvInternationalString          *psDocumentId;
DbvInternationalString          *psAbstract;
DbvOtherInformation            **ppsOtherInformation;
} DbvSummaryRecord;
```

2.2.14 Resource Report Record

```
/* -------------- Record Syntax "Resource Report" ----------------*/

typedef enum DbvEstimateType {
                 ET_CurrentSearchRecords    = 1,
                 ET_FinalSearchRecords      = 2,
                 ET_CurrentPresentRecords   = 3,
                 ET_FinalPresentRecords     = 4,
                 ET_CurrentOpTimeProcessing = 5,
                 ET_FinalOpTimeProcessing   = 6,
                 ET_CurrentAssocTime        = 7,
                 ET_CurrentOperationCost    = 8,
                 ET_FinalOperationCost      = 9,
                 ET_CurrentAssocCost        = 10,
                 ET_FinalOpTimeElapsed      = 11,
                 ET_PercentComplete         = 12,
                 ET_CurrentSearchAssocCost  = 13,
                 ET_CurrentPresentAssocCost = 14,
                 ET_CurrentConnectAssocCost = 15,
                 ET_CurrentOtherAssocCost   = 16,
                 ET_Invalid                 = -1
} DbvEstimateType;


typedef struct DbvEstimate1 {
DbvEstimateType          eEstimateType;
Ulong                    ulValue;
Ulong                   *pulCurrencyCode;
} DbvEstimate1;


typedef struct DbvResourceReport1 {
DbvEstimate1                **ppsEstimate;
DbvInternationalString        sMessage;
} DbvResourceReport1;


typedef struct DbvEstimate2 {
DbvStringOrNumeric       sType;
DbvIntUnit               sValue;
} DbvEstimate2;

typedef struct DbvResourceReport2 {
DbvEstimate2               **ppsEstimate;
```

```
DbvInternationalString      *psMessage;
} DbvResourceReport2;
```

2.2.15 Extended Services Task Package

```
/* -------------- Record Syntax "Extended Services Task Package" ------------*/

typedef enum DbvExtServicesTaskStatus {
                ESTS_Pending  = 1,
                ESTS_Active   = 2,
                ESTS_Complete = 3,
                ESTS_Aborted  = 4,
                ESTS_Invalid  = -1
} DbvExtServicesTaskStatus;


typedef struct DbvEsTaskPackage {
DbvObjectIdentifier                  pcPackageType;
DbvInternationalString              *psPackageName;
DbvInternationalString              *psUserId;
DbvIntUnit                          *psRetentionTime;
DbvPermissions                     **ppsPermissions;
DbvInternationalString              *psDescription;
DbvOctetString                      *psTargetReference;
DbvGeneralizedTime                   pcCreationDateTime;
DbvExtServicesTaskStatus             eExtServicesTaskStatus;
DbvDiagRec                         **ppsPackageDiagnostics;
DbvASN1External                      sTaskSpecificParameters;
} DbvEsTaskPackage;
```

2.2.16 Update Extended Services

```
/* -------------  UPDATE  EXTENDED SERVICE ---------------- */

typedef enum DbvUpdtAction {
                        UA_RecordInsert  = 1,
                        UA_RecordReplace = 2,
                        UA_RecordDelete  = 3,
                        UA_ElementUpdate = 4,
                        UA_Invalid       = -1
} DbvUpdtAction;
```

```
typedef enum DbvUpdtTaskStatus {
                 UTS_Success  = 1,
                 UTS_Partial  = 2,
                 UTS_Failure  = 3,
                 UTS_Invalid  = -1
} DbvUpdtTaskStatus;


typedef enum DbvUpdtRecordStatus {
                 URST_Success   = 1,
                 URST_Queued    = 2,
                 URST_InProcess = 3,
                 URST_Failure   = 4,
                 URST_Invalid   = -1
} DbvUpdtRecordStatus;


typedef enum DbvUpdtRecordOrDiagType {
                 URDT_Record     = 1,
                 URDT_Diagnostic = 2
} DbvUpdtRecordOrDiagType;


typedef enum DbvUpdtSupplementalIdType {
                USIT_Time            = 1,
                USIT_Version         = 2,
                USIT_PreviousVersion = 3
} DbvUpdtSupplementalIdType;


typedef enum DbvUpdtRecordIdType {
                URIT_Number = 1,
                URIT_String = 2,
                URIT_Opaque = 3
} DbvUpdtRecordIdType;


typedef struct DbvUpdtRecordOrDiag {
DbvUpdtRecordOrDiagType      eUpdtRecordOrDiag;
union {
    DbvASN1External      *psRecord;
    DbvDiagRec           *psDiagnostic;
  }                          uUpdtRecordOrDiag;
} DbvUpdtRecordOrDiag;


typedef struct DbvCorrelationInfo {
DbvInternationalString  *psNote;
Ulong                   *pulId;
} DbvCorrelationInfo;


typedef struct DbvTaskPackageRecord {
DbvUpdtRecordOrDiag       *psUpdtRecordOrDiag;
DbvCorrelationInfo        *psCorrelationInfo;
DbvUpdtRecordStatus        eUpdtRecordStatus;
} DbvTaskPackageRecord;
```

```
typedef struct DbvUpdateTargetPart {
DbvUpdtTaskStatus            eTaskStatus;
DbvDiagRec                 **ppsGlobalDiagnostics;
DbvTaskPackageRecord       **ppsTaskPackageRecord;
} DbvUpdateTargetPart;


typedef struct DbvUpdtSupplementalId {
DbvUpdtSupplementalIdType      eSupplementalIdType;
union {
    DbvGeneralizedTime       pcTime;
    DbvInternationalString *psVersion;
    DbvASN1External         *psPreviousVersion;
  }                            uUpdtSupplementalId;
} DbvUpdtSupplementalId;


typedef struct DbvUpdtRecordId {
DbvUpdtRecordIdType   eRecordIdType;
union {
    Ulong                    ulNumber;
    DbvInternationalString *psString;
    DbvOctetString          *psOpaque;
  }                   uUpdtRecordId;
} DbvUpdtRecordId;


typedef struct DbvSuppliedRecord {
DbvUpdtRecordId             *psUpdtRecordId;
DbvUpdtSupplementalId       *psUpdtSupplementalId;
DbvCorrelationInfo          *psCorrelationInfo;
DbvASN1External              sRecord;
} DbvSuppliedRecord;


typedef struct DbvOriginPartToKeep {
DbvUpdtAction                eAction;
DbvInternationalString       sDatabaseName;
DbvObjectIdentifier          pcSchema;
DbvInternationalString      *psElementSetName;
} DbvOriginPartToKeep;


typedef struct DbvUpdateTaskPackage {
DbvOriginPartToKeep       sOriginPartToKeep;
DbvUpdateTargetPart       sTargetPart;
} DbvUpdateTaskPackage;


typedef struct DbvUpdateRequestPackage {
DbvOriginPartToKeep    sOriginPartToKeep;
DbvSuppliedRecord    **ppsSuppliedRecord;
} DbvUpdateRequestPackage;


typedef enum DbvUpdateReqOrRsp {
                    UROR_Request  = 1,
```

```
                     UROR_Response = 2
} DbvUpdateReqOrRsp;


typedef struct DbvUpdateExtendedService {
DbvUpdateReqOrRsp     eUpdateReqRsp;
union {
    DbvUpdateRequestPackage  *psUpdateRequest;
    DbvUpdateTaskPackage     *psUpdateResponse;
  }                uUpdateReqOrRsp;
} DbvUpdateExtendedService;
```

2.3 Structures Associated with SR/Z39.50 Functions

```
/* Structures associated with SR-/Z39.50 functions */

/* For the SR-/Z39.50 service request and response structures.refer to the
annex. */


typedef struct DbvOriginData {
  DbvOriginEvent        eOriginEvent;
  union {
    DbvInitializeRsp              *psInitializeRsp;
    DbvSearchRsp                  *psSearchRsp;
    DbvPresentRsp                 *psPresentRsp;
    DbvDeleteResultSetRsp         *psDeleteResultSetRsp;
    DbvScanRsp                    *psScanRsp;
    DbvResourceReportRsp          *psResourceReportRsp;
    DbvSegmentReq                 *psSegmentReq;
    DbvSortRsp                    *psSortRsp;
    DbvCloseReq                   *psCloseReq;
    DbvExtendedServicesRsp        *psExtendedServicesRsp;
/*     DbvReleaseReq              *peReleaseReq;   Target is not allowed to
close the association */
    DbvOctetString                *psInvalidAPDU;

    struct DbvAccessControlReq                 *psAccessControlReq; /*
AccessControlRequest from target */
    struct DbvResourceControlReq         *psResourceControlReq; /*
ResourceControlRequest from target */


  }   uOriginData;
} DbvOriginData;

typedef struct DbvTargetData {
  DbvTargetEvent  eTargetEvent;
  union {
    DbvInitializeReq          *psInitializeReq;
    DbvSearchReq              *psSearchReq;
    DbvPresentReq             *psPresentReq;
    DbvDeleteResultSetReq     *psDeleteResultSetReq;
    DbvScanReq                *psScanReq;
    DbvResourceReportReq      *psResourceReportReq;
    DbvSortReq                *psSortReq;
    DbvCloseReq               *psCloseReq;
    DbvExtendedServicesReq    *psExtendedServicesReq;
    DbvReleaseReq             *peReleaseReq;
    DbvOctetString            *psInvalidAPDU;

    struct DbvAccessControlRsp                 *psAccessControlRsp; /*
AccessControlResponse from origin */
    struct DbvResourceControlRsp         *psResourceControlRsp; /*
ResourceControlResponse from origin */
    struct DbvTriggerResourceControlReq   *psTriggerResourceControlReq; /*
TriggerResourceControlResponse from origin */
```

```
  }              uTargetData;
} DbvTargetData;


#endif
```

3 Types and Sub-Structures in sitypes2.h

```
/* ----------  sitypes2.h -------------------------------------------*/


/* When using this file with other DBV-OSI II includes, include as follows

#include "sidefs.h"
#include "sitypes.h"
#include "sitypes2.h"
#include "dbvproto.h"
#include "iord_ext.h"
#include "itemordr.h"
*/
#ifdef PATCH
/*
#include "crxtypes.h"
*/
#endif /* divert(0) */



/*
   New Structure Definitions For ONE Extensions to DBV-OSI II
   ------------------------------------------------------------------

   Definitions contained in this file:


   New PDU Definitions
   ===================

   DbvAccessControlReq
   DbvAccessControlRsp
   DbvResourceControlReq
   DbvResourceControlRsp
   DbvTriggerResourceControlReq

   Notes:
   1) All optional fields are pointers.
   2) All choice enumerations start from 1.


   New External Definitions
   ========================

   AT_ExplainRecord
*/

#ifdef PATCH
/*
   AT_GenericRecord
*/
#endif /* divert(0) */
```

```
/*
   AT_PromptObject

*/
#ifdef PATCH
/*
   AT_AccessControlDes1
   AT_AccessControlKrb1

   AT_PerResSetExtendedService
   AT_PerQueryExtendedService
   AT_PerQuerySchExtendedService
   AT_ItemOrderExtendedService (in itemordr.h)
   AT_UpdateExtendedService (in sitypes.h)
   AT_ExpSpecExtendedService
   AT_ExpInvoExtendedService

   AT_UserInfoSearchResult

   AT_ElementSpec1
*/
#endif /* divert(0) */
/*
   Notes:
   1) All optional fields not in unions are pointers
   (fields which are pointers to chars however
   (pc) can be compulsory, depending on the field).
   2) All choice enumerations start from 1.
   3) Int enumerations are as defined in Z39.50 (some start from 0).
   4) Pointers to DbvNull structures are declared as DbvNull.
   5) DbvNull structures are not declared.
   6) All union members except enumerations, unsigned ints and bools are
   decalared as pointers (in keeping with sitypes.h style).
   7) Sequences are always represented as **, irrespective or being compulsory
   or optional.


   Structure ASN1External in sitypes.h has been updated to take account of
   the new types.
*/




/*  Module Name: @(#)sitypes2.h     7.11
    Version: 7/3/97
    Author: John Hough

    Crossnet Systems Ltd.
    Unit 3, The Paddock,
    Hambridge Road,
    Newbury,
    Berkshire, RG14 5TQ


    Description:
```

```
*/

/*
```

3.1 Additional Type Definitions

```
    Type definitions for ONE extensions to DBV-OSI II
*/

/*
    Note:
    Optional fields are pointers, compulsory fields are non-pointers.

    but also..

    Lists (optional or compulsory) are always pointers (**).
    Union members which are structures are specified as pointers.

*/

#if !defined(SI_TYPES2_H)
#define SI_TYPES2_H


/* ------------------------------------------------------------*/
/*                  Types and Sub-Structures                   */
/* ------------------------------------------------------------*/


/* -------------   DbvLanguageCode   ------------------------------ */

typedef DbvInternationalString DbvLanguageCode;


/* -------------   DbvNull   ------------------------------ */

typedef void* DbvNull;


/* -------------   DbvSecurityChallengeReq  ----------------------------- */

typedef enum DbvSecurityChallengeType {
                   SECT_SimpleForm       = 1,
                   SECT_ExternallyDefined= 2,
                   SECT_Invalid                = -1
} DbvSecurityChallengeType;



typedef struct DbvSecurityChallengeReq
{
DbvSecurityChallengeType   eSecurityChallengeReqType;
union {
```

```
    DbvOctetString              *psSimpleForm;
    DbvASN1External             *psExternallyDefinedInfo;
  }                  uSecurityChallengeReq;
} DbvSecurityChallengeReq;



/* -------------   DbvSecurityChallengeRsp   ----------------------------- */

typedef struct DbvSecurityChallengeRsp
{
DbvSecurityChallengeType   eSecurityChallengeRspType;
union {
    DbvOctetString              *psSimpleForm;
    DbvASN1External             *psExternallyDefinedInfo;
  }                  uSecurityChallengeRsp;
} DbvSecurityChallengeRsp;
```

3.1.1 Access Control Structure

```
/* ------------------    ACCESSCONTROL   --------------------------------- */

typedef struct DbvAccessControlReq
{
DbvOctetString               *psReferenceId;
DbvSecurityChallengeReq       sSecurityChallengeReq;
DbvOtherInformation         **ppsOtherInformation;
} DbvAccessControlReq;


typedef struct DbvAccessControlRsp
{
DbvOctetString               *psReferenceId;
DbvSecurityChallengeRsp      *psSecurityChallengeRsp;
DbvDiagRec                   *psDiagnostic;
DbvOtherInformation         **ppsOtherInformation;
} DbvAccessControlRsp;
```

3.1.2 Resource Control Structure

```
/* ------------------    RESOURCECONTROL   --------------------------------- */

typedef enum DbvPartialResultsAvailable {
              PRA_Subset   = 1,
              PRA_Interim2 = 2,
```

```
              PRA_None2    = 3,
              PRA_Invalid  = -1
} DbvPartialResultsAvailable;



typedef struct DbvResourceControlReq
{
DbvOctetString               *psReferenceId;
DbvBool                      *pfSuspendedFlag;
DbvASN1External              *psResourceReport;
DbvPartialResultsAvailable   *pePartialResultsAvailable;
DbvBool                       fResponseRequired;
DbvBool                      *pfTriggeredRequestFlag;
DbvOtherInformation         **ppsOtherInformation;
} DbvResourceControlReq;



typedef struct DbvResourceControlRsp
{
DbvOctetString               *psReferenceId;
DbvBool                       fContinueFlag;
DbvBool                      *pfResultSetWanted;
DbvOtherInformation         **ppsOtherInformation;
} DbvResourceControlRsp;
```

3.1.3 Trigger Access Control Structure

```
/* -----------    TRIGGERRESOURCECONTROL   -------------------------- */


typedef enum DbvRequestedAction {
              RA_ResourceReport1   = 1,
              RA_ResourceControl   = 2,
              RA_Cancel            = 3,
              RA_Invalid           = -1
} DbvRequestedAction;



typedef struct DbvTriggerResourceControlReq
{
DbvOctetString               *psReferenceId;
DbvRequestedAction            eRequestedAction;
DbvObjectIdentifier           pcPreferredResourceReportFormat;
DbvBool                      *pfResultSetWanted;
DbvOtherInformation         **ppsOtherInformation;
} DbvTriggerResourceControlReq;


/* TriggerResourceControlRequest is sent from origin to target.
   There is no TriggerResourceControlResponse in Z39.50 V3 */
```

3.2 Additional Record Syntaxes

```
/* -----------------------------------------------------------*/
/*    Appendix REC Record Syntaxes                            */
/* -----------------------------------------------------------*/
```

3.2.1 Explain Record Syntax

```
/* -------------- Record Syntax "DbvExplainRecord" ----------------*/


typedef enum DbvExplainRecordType {
                   ERT_TargetInfo              = 1,
                   ERT_DatabaseInfo       = 2,
                   ERT_SchemaInfo         = 3,
                   ERT_TagSetInfo         = 4,
                   ERT_RecordSyntaxInfo         = 5,
                   ERT_AttributeSetInfo         = 6,
                   ERT_TermListInfo       = 7,
                   ERT_ExtendedServicesInfo    = 8,
                   ERT_AttributeDetails         = 9,
                   ERT_TermListDetails          = 10,
                   ERT_ElementSetDetails       = 11,
                   ERT_RetrievalRecordDetails  = 12,
                   ERT_SortDetails             = 13,
                   ERT_Processing              = 14,
                   ERT_Variants                = 15,
                   ERT_Units                   = 16,
                   ERT_CategoryList       = 17,
                   ERT_Invalid                 = -1
} DbvExplainRecordType;


/* -------------- DbvTargetInfo ----------------*/

typedef struct DbvCommonInfo
{
DbvGeneralizedTime            pcDateAdded;
DbvGeneralizedTime            pcDateChanged;
DbvGeneralizedTime            pcExpiry;
DbvLanguageCode              *psHumanStringLanguage;
DbvOtherInformation         **ppsOtherInformation;
} DbvCommonInfo;



typedef struct DbvHumanString
{
```

```
DbvLanguageCode         *psLanguage;
DbvInternationalString   sText;
} DbvHumanString;



typedef enum DbvBodyTypeType {
                   BTT_IanaType          = 1,
                   BTT_Z3950Type    = 2,
                   BTT_OtherType    = 3,
                   BTT_Invalid          = -1
} DbvBodyTypeType;



typedef struct DbvBodyType
{
DbvBodyTypeType   eBodyTypeType;
union {
    DbvInternationalString      *psIanaType;
    DbvInternationalString      *psZ3950Type;
    DbvInternationalString      *psOtherType;
  }                  uBodyType;
} DbvBodyType;



typedef struct DbvIconObject
{
DbvBodyType       sBodyType;
DbvOctetString    sContent;
} DbvIconObject;


typedef struct DbvContactInfo
{
DbvInternationalString    *psName;
DbvHumanString           **ppsDescription;
DbvHumanString           **ppsAddress;
DbvInternationalString    *psEmail;
DbvInternationalString    *psPhone;
} DbvContactInfo;


typedef struct DbvDatabaseList
{
DbvInternationalString **ppsInnerDatabases; /* mand. */
} DbvDatabaseList;



typedef enum DbvNetworkAddressType {
                   NAT_InternetAddress          = 1,
                   NAT_OsiPresentationAddress   = 2,
                   NAT_OtherAddress             = 3,
                   NAT_Invalid                  = -1
} DbvNetworkAddressType;
```

```
typedef struct DbvInterAddress
{
DbvInternationalString    sHostAddress;
Ulong                     ulPort;
} DbvInterAddress;



typedef struct DbvOsiPresentationAddress
{
DbvInternationalString    sPsel;
DbvInternationalString   *psSSel;
DbvInternationalString   *psTSel;
DbvInternationalString    sNSap;
} DbvOsiPresentationAddress;



typedef struct DbvOtherAddress
{
DbvInternationalString    sType;
DbvInternationalString    sAddress;
} DbvOtherAddress;



typedef struct DbvNetworkAddress
{
DbvNetworkAddressType   eNetworkAddressType;
union {
    DbvInterAddress                *psInternetAddress;
    DbvOsiPresentationAddress      *psOsiPresentationAddress;
    DbvOtherAddress                *psOtherAddress;
  }                   uNetworkAddress;
} DbvNetworkAddress;




typedef enum DbvQueryTypeDetailsType {
         QTDT_Private     = 1,
         QTDT_Rpn         = 2,
         QTDT_Iso8777     = 3,
         QTDT_Z39_58      = 4,
         QTDT_Erpn        = 5,
         QTDT_RankedList  = 6,
         QTDT_Invalid     = -1
} DbvQueryTypeDetailsType;



typedef struct DbvOperators
{
DbvInternationalString    sOperator;
DbvHumanString          **ppsDescription;
} DbvOperators;
```

```
typedef struct DbvSearchKey
{
DbvInternationalString    sSearchKey;
DbvHumanString          **ppsDescription;
} DbvSearchKey;



typedef struct DbvHumanStringSeq
{
DbvHumanString     **ppsHumanString;
} DbvHumanStringSeq;



typedef struct DbvPrivateCapabilities
{
DbvOperators       **ppsOperators;
DbvSearchKey       **ppsSearchKeys;
DbvHumanStringSeq  **ppsDescription;
} DbvPrivateCapabilities;




typedef enum DbvUnitsSupportedType {
       UST_Known   = 1,
       UST_Private = 2,
       UST_Invalid = -1
} DbvUnitsSupportedType;


typedef struct DbvPrivate
{
Ulong          ulUnit;
DbvHumanString **ppsDescription;
} DbvPrivate;



typedef struct DbvUnitsSupported
{
DbvUnitsSupportedType eUnitsSupportedType;
union {
   Ulong        ulKnown;
   DbvPrivate *psPrivate;
  } uUnitsSupported;
} DbvUnitsSupported;



typedef struct DbvProximitySupport
{
DbvBool              fAnySupport;
DbvUnitsSupported  **ppsUnitsSupported;
```

```
} DbvProximitySupport;



typedef struct DbvRpnCapabilities
{
Ulong              **ppulOperators;
DbvBool              fResultSetAsOperandSupported;
DbvBool              fRestrictionOperandSupported;
DbvProximitySupport *psProximity;
} DbvRpnCapabilities;



typedef struct DbvIso8777Capabilities
{
DbvSearchKey     **ppsSearchKeys;
DbvHumanString   **ppsRestrictions;
} DbvIso8777Capabilities;



typedef struct DbvQueryTypeDetails
{
DbvQueryTypeDetailsType     eQueryTypeDetailsType;
union {
     DbvPrivateCapabilities  *psPrivate;
     DbvRpnCapabilities      *psRpn;
     DbvIso8777Capabilities  *psIso8777;
     DbvHumanString         **ppsZ39_58;
     DbvRpnCapabilities      *psErpn;
     DbvHumanString         **ppsRankedList;
  } uQueryTypeDetails;
} DbvQueryTypeDetails;



typedef enum DbvAccessType {
    AT_Any               = 0,
    AT_Search            = 1,
    AT_Present           = 2,
    AT_SpecificElements = 3,
    AT_ExtendedServices = 4,
    AT_ByDatabase        = 5,
    AT_Invalid           = -1
} DbvAccessType;



typedef struct DbvAccessRestrictions
{
DbvAccessType             eAccessType;
DbvHumanString          **ppsAccessText;
DbvObjectIdentifier      *ppcAccessChallenges;
} DbvAccessRestrictions;
```

```
typedef struct DbvCharge
{
DbvIntUnit       sCost;
DbvUnit         *psPerWhat;
DbvHumanString **ppsText;
} DbvCharge;



typedef struct DbvOtherCharge
{
DbvHumanString **ppsForWhat;
DbvCharge        sCharge;
} DbvOtherCharge;



typedef struct DbvCosts
{
DbvCharge       *psConnectCharge;
DbvCharge       *psConnectTime;
DbvCharge       *psDisplayCharge;
DbvCharge       *psSearchCharge;
DbvCharge       *psSubscriptCharge;
DbvOtherCharge **ppsOtherCharge;
} DbvCosts;


/* A ** is not needed for ppcDiagnosticsSets etc. because
   DbvObjectIdentifier is a typedefed char * (see sitypes.h) */

typedef struct DbvAccessInfo
{
DbvQueryTypeDetails           **ppsQueryTypesSupported;
DbvObjectIdentifier            *ppcDiagnosticsSets;
DbvObjectIdentifier            *ppcAttributeSetIds;
DbvObjectIdentifier            *ppcSchemas;
DbvObjectIdentifier            *ppcRecordSyntaxes;
DbvObjectIdentifier            *ppcResourceChallenges;
DbvAccessRestrictions         **ppsRestrictedAccess;
DbvCosts                       *psCostInfo;
DbvObjectIdentifier            *ppcVariantSets;
DbvInternationalString        **ppsElementSetNames;
DbvInternationalString        **ppsUnitSystems;
} DbvAccessInfo;



typedef struct DbvTargetInfo
{
DbvCommonInfo             *psCommonInfo;
DbvInternationalString     sName;
DbvHumanString           **ppsRecentNews;
DbvIconObject            **ppsIcon;
DbvBool                    fNamedResultSets;
DbvBool                    fMultipleDBSearch;
Ulong                     *pulMaxResultSets;
Ulong                     *pulMaxResultSize;
```

```
Ulong                        *pulMaxTerms;
DbvIntUnit                   *psTimeoutInterval;
DbvHumanString              **ppsWelcomeMessage;
DbvContactInfo               *psContactInfo;
DbvHumanString              **ppsDescription;
DbvInternationalString      **ppsNicknames;
DbvHumanString              **ppsUsageRest;
DbvHumanString              **ppsPaymentAddr;
DbvHumanString              **ppsHours;
DbvDatabaseList             **ppsDbCombinations;
DbvNetworkAddress           **ppsAddresses;
DbvInternationalString      **ppsLanguages;
DbvAccessInfo                *psCommonAccessInfo;
} DbvTargetInfo;


/* -------------- DbvDatabaseInfo ----------------*/

typedef enum DbvRecordCountType {
     RCT_ActualNumber = 1,
     RCT_ApproxNumber = 2,
     RCT_Invalid      = -1
} DbvRecordCountType;


typedef struct DbvRecordCount
{
DbvRecordCountType eRecordCountType;
union {
     Ulong ulActualNumber;
     Ulong ulApproxNumber;
  } uRecordCount;
} DbvRecordCount;


typedef struct DbvDataBaseInfoSeq
{
DbvHumanString              **ppsDataBaseInfo;
} DbvDataBaseInfoSeq;


typedef struct DbvDatabaseInfo
{
DbvCommonInfo                *psCommonInfo;
DbvInternationalString        sName;
DbvNull                       pvExplainDatabase; /* Set to either NULL or
                                  1 alloced byte */
DbvInternationalString      **ppsNicknames;
DbvIconObject               **ppsIcon;
DbvBool                       fUserFee;
DbvBool                       fAvailable;
DbvHumanString              **ppsTitleString;
DbvDataBaseInfoSeq          **ppsKeywords;
DbvHumanString              **ppsDescription;
DbvInternationalString      **ppsAssociatedDbs;
DbvInternationalString      **ppsSubDbs;
DbvHumanString              **ppsDisclaimers;
DbvHumanString              **ppsNews;
```

```
DbvRecordCount            *psRecordCount;
DbvHumanString           **ppsDefaultOrder;
Ulong                     *pulAvRecordSize;
Ulong                     *pulMaxRecordSize;
DbvHumanString           **ppsHours;
DbvHumanString           **ppsBestTime;
DbvGeneralizedTime         pcLastUpdate;
DbvIntUnit                *psUpdateInterval;
DbvHumanString           **ppsCoverage;
DbvBool                   *pfProprietary;
DbvHumanString           **ppsCopyrightText;
DbvHumanString           **ppsCopyrightNotice;
DbvContactInfo            *psProducerContactInfo;
DbvContactInfo            *psSupplierContactInfo;
DbvContactInfo            *psSubmissionContactInfo;
DbvAccessInfo             *psAccessInfo;
} DbvDatabaseInfo;
```

/* -------------- DbvSchemaInfo ----------------*/

```
typedef struct DbvTagTypeMapping {
Ulong                   ulTagType;
DbvObjectIdentifier     pcTagSet;
DbvNull                 pvDefaultTagType; /* Set to either NULL or
                                   1 alloced byte */
} DbvTagTypeMapping;



typedef struct DbvPath {
Ulong              ulTagType;
DbvStringOrNumeric sTagValue;
} DbvPath;



typedef struct DbvElementInfo {
DbvInternationalString      sElementName;
DbvPath                   **ppsElementTagPath;
struct DbvElementDataType   *psDataType;
DbvBool                      fRequired;
DbvBool                      fRepeatable;
DbvHumanString             **ppsDescription;
} DbvElementInfo;



typedef enum DbvElementDataTypeType {
   EDTT_Primitive  = 1,
   EDTT_Structured = 2,
   EDTT_Invalid    = -1
} DbvElementDataTypeType;



typedef enum DbvPrimitiveDataType {
   PDT_OctetString     = 0,
```

```
   PDT_Numeric         = 1,
   PDT_Date            = 2,
   PDT_External        = 3,
   PDT_String          = 4,
   PDT_TrueOrFalse     = 5,
   PDT_Oid             = 6,
   PDT_IntUnit         = 7,
   PDT_Empty           = 8,
   PDT_NoneOfTheAbove  = 100,
   PDT_Invalid         = -1
} DbvPrimitiveDataType;



typedef struct DbvElementDataType {
DbvElementDataTypeType eElementDataTypeType;
union {
     DbvPrimitiveDataType       ePrimitive;
     DbvElementInfo           **ppsStructured;
  } uElementDataType;
} DbvElementDataType;



typedef struct DbvSchemaInfo {
DbvCommonInfo            *psCommonInfo;
DbvObjectIdentifier       pcSchema;
DbvInternationalString    sName;
DbvHumanString          **ppsDescription;
DbvTagTypeMapping       **ppsTagTypeMapping;
DbvElementInfo          **ppsRecordStructure;
} DbvSchemaInfo;


/* -------------- DbvTagSetInfo ----------------*/

typedef struct DbvElements
{
DbvInternationalString    sElementName;
DbvInternationalString  **ppsNicknames;
DbvStringOrNumeric        sElementTag;
DbvHumanString          **ppsDescription;
DbvPrimitiveDataType     *peDataType;
DbvOtherInformation     **ppsOtherInformation;
} DbvElements;



typedef struct DbvTagSetInfo
{
DbvCommonInfo            *psCommonInfo;
DbvObjectIdentifier       pcTagSet;
DbvInternationalString    sName;
DbvHumanString          **ppsDescription;
DbvElements             **ppsElements;
} DbvTagSetInfo;
```

```
/* -------------- DbvRecordSyntaxInfo ----------------*/

typedef struct DbvRecordSyntaxInfo
{
DbvCommonInfo            *psCommonInfo;
DbvObjectIdentifier       pcRecordSyntax;
DbvInternationalString    sName;
DbvObjectIdentifier      *ppcTransferSyntaxes;
DbvHumanString          **ppsDescription;
DbvInternationalString   *psAsn1Module;
DbvElementInfo          **ppsAbstractStructure;
} DbvRecordSyntaxInfo;


/* -------------- DbvAttributeSetInfo ----------------*/

typedef struct DbvAttributeDescription
{
DbvInternationalString   *psName;
DbvHumanString          **ppsDescription;
DbvStringOrNumeric        sAttributeValue;
DbvStringOrNumeric      **ppsEquivalentAttributes;
} DbvAttributeDescription;



typedef struct DbvAttributeType
{
DbvInternationalString     *psName;
DbvHumanString            **ppsDescription;
Ulong                       ulAttributeType;
DbvAttributeDescription **ppsAttributeValues;
} DbvAttributeType;


typedef struct DbvAttributeSetInfo
{
DbvCommonInfo                *psCommonInfo;
DbvObjectIdentifier           pcAttributeSetId;
DbvInternationalString        sName;
DbvAttributeType            **ppsAttributes;
DbvHumanString              **ppsDescription;
} DbvAttributeSetInfo;



/* -------------- DbvTermListInfo ----------------*/

typedef enum DbvSearchCost {
   SC_Optimized = 0,
   SC_Normal    = 1,
   SC_Expensive = 2,
   SC_Filter    = 3,
   SC_Invalid   = -1
} DbvSearchCost;
```

```
typedef struct DbvTermList
{
DbvInternationalString    sName;
DbvHumanString          **ppsTitle;
DbvSearchCost            *peSearchCost;
DbvBool                   fScanable;
DbvInternationalString  **ppsBroader;
DbvInternationalString  **ppsNarrower;
} DbvTermList;



typedef struct DbvTermListInfo
{
DbvCommonInfo           *psCommonInfo;
DbvInternationalString  sDatabaseName;
DbvTermList           **ppsTermLists;
} DbvTermListInfo;


/* -------------- DbvExtendedServicesInfo ----------------*/

typedef enum DbvExtendedServicesInfoInt
{
   ESII_WaitSupported    = 1,
   ESII_WaitAlways       = 2,
   ESII_WaitNotSupported = 3,
   ESII_Depends          = 4,
   ESII_NotSaying        = 5,
   ESII_Invalid          = -1
} DbvExtendedServicesInfoInt;



typedef struct DbvExtendedServicesInfo
{
DbvCommonInfo                *psCommonInfo;
DbvObjectIdentifier           pcType;
DbvInternationalString       *psName;
DbvBool                       fPrivateType;
DbvBool                       fRestrictionsApply;
DbvBool                       fFeeApply;
DbvBool                       fAvailable;
DbvBool                       fRetentionSupported;
DbvExtendedServicesInfoInt    eWaitAction;
DbvHumanString              **ppsDescription;
DbvASN1External              *psSpecificExplain;
DbvInternationalString       *psEsASN;
} DbvExtendedServicesInfo;


/* -------------- DbvAttributeDetails ----------------*/

typedef struct DbvOmittedAttributeInterpretation
{
DbvStringOrNumeric  *psDefaultValue;
DbvHumanString     **ppsDefaultDescription;
} DbvOmittedAttributeInterpretation;
```

```
typedef struct DbvAttributeValues
{
DbvStringOrNumeric    sValue;
DbvHumanString      **ppsDescription;
DbvStringOrNumeric  **ppsSubAttributes;
DbvStringOrNumeric  **ppsSuperAttributes;
DbvNull               pvPartialSupport; /* Set to either NULL or
                                 1 alloced byte */
} DbvAttributeValues;



typedef struct DbvAttributesByType
{
Ulong                              ulAttributeType;
DbvOmittedAttributeInterpretation *psDefaultIfOmitted;
DbvAttributeValues               **ppsAttributeValues;
} DbvAttributesByType;


typedef struct DbvAttributesBySet
{
DbvObjectIdentifier       pcAttributeSet;
DbvAttributesByType     **ppsAttributesByType;
} DbvAttributesBySet;


typedef enum DbvAttributeOccurrenceChoiceType {
    AOT_AnyOrNone             = 1,
    AOT_Specific              = 2,
    AOT_Invalid               = -1
} DbvAttributeOccurrenceChoiceType;



typedef struct DbvAttributeOccurrenceChoice
{
DbvAttributeOccurrenceChoiceType   eAttributeOccurrenceChoiceType;
union {
     DbvNull               pvAnyOrNone;  /* mand - set to 1 alloced byte */
     DbvStringOrNumeric  **ppsSpecific;
  } uAttributeOccurrenceChoiceType;
} DbvAttributeOccurrenceChoice;



typedef struct DbvAttributeOccurrence
{
DbvObjectIdentifier                pcAttributeSet;
Ulong                              ulAttributeType;
DbvNull                            pvMustBeSupplied; /* Set to either NULL or
                                         1 alloced byte */
DbvAttributeOccurrenceChoice       sAttributeValues;
} DbvAttributeOccurrence;
```

```
/* Note that the following typedef is DbvAttributeCombination, typedef
   DbvAttributeCombinations (which is different [extra 's' at the end]
   follows later */

typedef struct DbvAttributeCombination
{
DbvAttributeOccurrence           **ppsAttributeOccurrence;
} DbvAttributeCombination;


typedef struct DbvAttributeCombinations
{
DbvObjectIdentifier                pcDefaultAttributeSet;
DbvAttributeCombination          **ppsLegalCombinations;
} DbvAttributeCombinations;



typedef struct DbvAttributeDetails
{
DbvCommonInfo            *psCommonInfo;
DbvInternationalString    sDatabaseName;
DbvAttributesBySet      **ppsAttributesBySet;
DbvAttributeCombinations *psAttributeCombinations;
} DbvAttributeDetails;

/* -------------- DbvTermListDetails ----------------*/

typedef struct DbvScanInfo
{
Ulong           *pulMaxStepSize;
DbvHumanString **ppsCollatingSequence;
DbvBool         *pfIncreasing;
} DbvScanInfo;



typedef struct DbvTermListDetails
{
DbvCommonInfo              *psCommonInfo;
DbvInternationalString      sTermListName;
DbvHumanString            **ppsDescription;
DbvAttributeCombinations   *psAttributes;
DbvScanInfo                *psScanInfo;
Ulong                      *pulEstNumberTerms;
DbvTerm                   **ppsSampleTerms;
} DbvTermListDetails;


/* -------------- DbvElementSetDetails ----------------*/

typedef struct DbvRecordTag
{
DbvStringOrNumeric *psQualifier;
DbvStringOrNumeric  sTagValue;
```

```
} DbvRecordTag;


typedef struct DbvPathA {
DbvPath                 **ppsSchemaTags;
} DbvPathA;

typedef struct DbvDetailsPerElement
{
DbvInternationalString    *psName;
DbvRecordTag              *psRecordTag;
DbvPathA                 **ppsSchemaTags;
Ulong                     *pulMaxSize;
Ulong                     *pulMinSize;
Ulong                     *pulAvgSize;
Ulong                     *pulFixedSize;
DbvBool                    fRepeatable;
DbvBool                    fRequired;
DbvHumanString           **ppsDescription;
DbvHumanString           **ppsContents;
DbvHumanString           **ppsBillingInfo;
DbvHumanString           **ppsRestrictions;
DbvInternationalString   **ppsAlternateNames;
DbvInternationalString   **ppsGenericNames;
DbvAttributeCombinations  *psSearchAccess;
} DbvDetailsPerElement;



typedef struct DbvElementSetDetails
{
DbvCommonInfo            *psCommonInfo;
DbvInternationalString    sDatabaseName;
DbvInternationalString    sElementSetName;
DbvObjectIdentifier       pcRecordSyntax;
DbvObjectIdentifier       pcSchema;
DbvHumanString          **ppsDescription;
DbvDetailsPerElement    **ppsDetailsPerElement;
} DbvElementSetDetails;



/* -------------- DbvRetrievalRecordDetails ----------------*/

typedef struct DbvRetrievalRecordDetails
{
DbvCommonInfo           *psCommonInfo;
DbvInternationalString   sDatabaseName;
DbvObjectIdentifier      pcSchema;
DbvObjectIdentifier      pcRecordSyntax;
DbvHumanString         **ppsDescription;
DbvDetailsPerElement   **ppsDetailsPerElement;
} DbvRetrievalRecordDetails;


/* -------------- DbvSortDetails ----------------*/

typedef enum DbvSortTypeType {
```

```
     STT_Character  = 1,
     STT_Numeric    = 2,
     STT_Structured = 3,
     STT_Invalid    = -1
} DbvSortTypeType;



typedef struct DbvSortType
{
DbvSortTypeType       eSortTypeType;
union {
    DbvHumanString **ppsStructured; /* Set if eSortTypeType == STT_Structured */
     } uSortType;
} DbvSortType;



typedef enum DbvCaseSensitivity
{
    CSE_Always     = 0,
    CSE_Never      = 1,
    CSE_DefaultYes = 2,
    CSE_DefaultNo  = 3,
    CSE_Invalid    = -1
} DbvCaseSensitivity;



typedef struct DbvSortKeys
{
DbvHumanString           **ppsDescription;
DbvSpecification         **ppsElementSpecifications;
DbvAttributeCombinations  *psAttributeSpecifications;
DbvSortType               *psSortType;
DbvCaseSensitivity        *peCaseSensitivity;
} DbvSortKeys;


typedef struct DbvSortDetails
{
DbvCommonInfo            *psCommonInfo;
DbvInternationalString    sDatabaseName;
DbvSortKeys            **ppsSortKeys;
} DbvSortDetails;


/* -------------- DbvProcessingInformation ----------------*/

typedef enum DbvProcessingContext
{
    PC_Access              = 0,
    PC_Search              = 1,
    PC_Retrieval           = 2,
    PC_RecordPresentation = 3,
    PC_RecordHandling      = 4,
    PC_Invalid             = -1
} DbvProcessingContext;
```

```
typedef struct DbvProcessingInformation
{
DbvCommonInfo             *psCommonInfo;
DbvInternationalString     sDatabaseName;
DbvProcessingContext       eProcessingContext;
DbvInternationalString     sName;
DbvObjectIdentifier        pcOid;
DbvHumanString           **ppsDescription;
DbvASN1External           *psInstructions;
} DbvProcessingInformation;
```

```
/* -------------- DbvVariantSetInfo ----------------*/

typedef enum DbvValueSetType {
     VT_Range      = 1,
     VT_Enumerated = 2,
     VT_Invalid    = -1
} DbvValueSetType;


typedef enum DbvValueDescriptionType {
    VDT_Integer       = 1,
    VDT_String        = 2,
    VDT_Octets        = 3,
    VDT_Oid           = 4,
    VDT_Unit          = 5,
    VDT_ValueAndUnit  = 6,
    VDT_Invalid       = -1
} DbvValueDescriptionType;



typedef struct DbvValueDescription
{
DbvValueDescriptionType eValueDescriptionType;
union {
     Ulong                   ulInteger;
     DbvInternationalString *psString;
     DbvOctetString         *psOctets;
     DbvObjectIdentifier     pcOid;
     DbvUnit                *psUnit;
     DbvIntUnit             *psValueAndUnit;
 } uValueDescription;
} DbvValueDescription;



typedef struct DbvValueRange
{
DbvValueDescription *psLower;
DbvValueDescription *psUpper;
} DbvValueRange;
```

```
typedef struct DbvValueSet
{
DbvValueSetType   eValueSetType;
union {
     DbvValueRange         *psRange;
     DbvValueDescription  **ppsEnumerated;
     } uValueSet;
} DbvValueSet;



typedef struct DbvVariantValue
{
DbvPrimitiveDataType  eDataType;
DbvValueSet          *psValues;
} DbvVariantValue;



typedef struct DbvVariantTypes
{
DbvInternationalString   *psName;
DbvHumanString          **ppsDescription;
Ulong                      ulVariantType;
DbvVariantValue          *psVariantValue;
} DbvVariantTypes;



typedef struct DbvVariantClass
{
DbvInternationalString  *psName;
DbvHumanString         **ppsDescription;
Ulong                    ulVariantClass;
DbvVariantTypes        **ppsVariantTypes;
} DbvVariantClass;



typedef struct DbvVariantSetInfo
{
DbvCommonInfo           *psCommonInfo;
DbvObjectIdentifier      pcVariantSet;
DbvInternationalString   sName;
DbvVariantClass        **ppsVariants;
} DbvVariantSetInfo;
```

/* -------------- DbvUnitInfo ----------------*/

```
typedef struct DbvUnits
{
DbvInternationalString  *psName;
DbvHumanString         **ppsDescription;
DbvStringOrNumeric       sUnit;
} DbvUnits;
```

```
typedef struct DbvUnitType
{
DbvInternationalString  *psName;
DbvHumanString         **ppsDescription;
DbvStringOrNumeric       sUnitType;
DbvUnits               **ppsUnits;
} DbvUnitType;


typedef struct DbvUnitInfo
{
DbvCommonInfo            *psCommonInfo;
DbvInternationalString    sUnitSystem;
DbvHumanString          **ppsDescription;
DbvUnitType             **ppsUnitTypes;
} DbvUnitInfo;


/* -------------- DbvCategoryList ----------------*/

typedef struct DbvCategoryInfo
{
DbvInternationalString   sCategory;
DbvInternationalString  *psOriginalCategory;
DbvHumanString         **ppsDescription;
DbvInternationalString  *psAsn1Module;
} DbvCategoryInfo;


typedef struct DbvCategoryList
{
DbvCommonInfo     *psCommonInfo;
DbvCategoryInfo  **ppsCategories;
} DbvCategoryList;



/* -------------------------------------------------------------*/

typedef struct DbvExplainRecord
{
DbvExplainRecordType   eExplainRecordType;
union {
    DbvTargetInfo               *psTargetInfo;
    DbvDatabaseInfo             *psDatabaseInfo;
    DbvSchemaInfo               *psSchemaInfo;
    DbvTagSetInfo               *psTagSetInfo;
    DbvRecordSyntaxInfo         *psRecordSyntaxInfo;
    DbvAttributeSetInfo         *psAttributeSetInfo;
    DbvTermListInfo             *psTermListInfo;
    DbvExtendedServicesInfo     *psExtendedServicesInfo;
    DbvAttributeDetails         *psAttributeDetails;
    DbvTermListDetails          *psTermListDetails;
    DbvElementSetDetails        *psElementSetDetails;
    DbvRetrievalRecordDetails   *psRetrievalRecordDetails;
    DbvSortDetails              *psSortDetails;
    DbvProcessingInformation    *psProcessing;
```

```
    DbvVariantSetInfo            *psVariants;
    DbvUnitInfo                  *psUnits;
    DbvCategoryList              *psCategoryList;
  }                  uExplainRecord;
} DbvExplainRecord;
```

3.2.2 GRS.1 Record Syntax

```
#ifdef PATCH

/* -------------- Record Syntax "GenericRecord" ----------------*/

typedef enum DbvElementDataGType {
    EDT_Octets             = 1,
    EDT_Numeric            = 2,
    EDT_Date               = 3,
    EDT_Ext                = 4,
    EDT_String             = 5,
    EDT_TrueOrFalse        = 6,
    EDT_Oid                = 7,
    EDT_IntUnit            = 8,
    EDT_ElementNotThere    = 9,
    EDT_ElementEmpty       = 10,
    EDT_NoDataRequested    = 11,
    EDT_Diagnostic         = 12,
    EDT_Subtree            = 13,
    EDT_Invalid            = -1
} DbvElementDataGType;


typedef struct DbvOrder
{
DbvBool fAscending;
Ulong   ulOrder;
} DbvOrder;


typedef enum DbvUsageIntType {
   UIT_Redistributable = 1,
   UIT_Restricted      = 2,
   UIT_LicensePointer  = 3,
   UIT_Invalid         = -1
} DbvUsageIntType;



typedef struct DbvUsage
{
DbvUsageIntType         eType;
DbvInternationalString *psRestriction;
} DbvUsage;
```

```
typedef struct DbvHits
{
DbvTerm        *psSatisfier;
DbvIntUnit     *psOffsetIntoElement;
DbvIntUnit     *psLength;
Ulong          *pulHitRank;
DbvOctetString *psTargetToken;
} DbvHits;


typedef enum DbvVariantValueType {
    VVT_Int                 = 1,
    VVT_InternationalString = 2,
    VVT_Octs                = 3,
    VVT_Oid                 = 4,
    VVT_Bool                = 5,
    VVT_Null                = 6,
    VVT_Unit                = 7,
    VVT_ValueAndUnit        = 8,
    VVT_Invalid             = -1
} DbvVariantValueType;


typedef struct DbvValue
{
DbvVariantValueType eVariantValueType;
union {
     Ulong                    ulInt1;
     DbvInternationalString *psInternationalString;
     DbvOctetString         *psOcts;
     DbvObjectIdentifier     pcOid;
     DbvBool                 fBool;
     DbvUnit                *psUnit;
     DbvIntUnit             *psValueAndUnit;
   } uVariantValueType;
} DbvValue;



typedef struct DbvTriples
{
DbvObjectIdentifier  pcVariantSetId;
Ulong                ulClass;
Ulong                ulType;
DbvValue             sValue;
} DbvTriples;



typedef struct DbvVariant
{
DbvObjectIdentifier  pcGlobalVariantSetId;
DbvTriples         **ppsTriples;
} DbvVariant;
```

```
typedef struct DbvTagPath
{
Ulong                *pulTagType;
DbvStringOrNumeric    sTagValue;
Ulong                *pulTagOccurrence;
} DbvTagPath;



typedef struct DbvElementMetaData
{
DbvOrder                 *psSeriesOrder;
DbvUsage                 *psUsageRight;
DbvHits                 **ppsHits;
DbvInternationalString   *psDisplayName;
DbvVariant              **ppsSupportedVariants;
DbvInternationalString   *psMessage;
DbvOctetString           *psElementDescriptor;
DbvTagPath              **ppsSurrogateFor;
DbvTagPath              **ppsSurrogateElement;
DbvASN1External          *psOther;
} DbvElementMetaData;




typedef struct DbvElementDataG
{
DbvElementDataGType   eElementDataGType;
union {
     DbvOctetString             *psOctets;
     Ulong                       ulNumeric;
     DbvGeneralizedTime          pcDate;
     DbvASN1External            *psExt;
     DbvInternationalString     *psString;
     DbvBool                     fTrueOrFalse;
     DbvObjectIdentifier         pcOid;
     DbvIntUnit                 *psIntUnit;
     DbvASN1External            *psDiagnostic;
     struct DbvTaggedElement   **ppsSubTree;
   } uElementDataG;
} DbvElementDataG;



typedef struct DbvTaggedElement
{
Ulong                    *pulTagType;
DbvStringOrNumeric        sTagValue;
Ulong                    *pulTagOccurrence;
DbvElementDataG           sContent;
DbvElementMetaData       *psMetaData;
DbvVariant               *psAppliedVariant;
} DbvTaggedElement;


typedef struct DbvGenericRecord
```

```
{
DbvTaggedElement        **ppsTaggedElement;
} DbvGenericRecord;


#endif /* divert(0) */
```

3.2.3 Access Control – Prompt-1 Record Syntax

```
/* -----------------------------------------------------------*/
/*    Appendix ACC Access Control Formats                     */
/* -----------------------------------------------------------*/


/* -------------- Access Control Format "Prompt 1" ----------------*/


typedef enum DbvPromptObjectType {
    POT_Challenge = 1,
    POT_Response  = 2,
    POT_Invalid   = -1
} DbvPromptObjectType;



typedef enum DbvPromptIdType {
    PIT_EnumeratedPrompt     = 1,
    PIT_NoneEnumeratedPrompt = 2,
    PIT_Invalid              = -1
} DbvPromptIdType;



typedef enum DbvEPromptIdType {
    EPIT_GroupId     = 0,
    EPIT_UserId      = 1,
    EPIT_Password    = 2,
    EPIT_NewPassword = 3,
    EPIT_Copyright   = 4,
    EPIT_SessionId   = 5,
    EPIT_Invalid     = -1
} DbvEPromptIdType;


typedef struct DbvEnumeratedPrompt
{
DbvEPromptIdType         eType;
DbvInternationalString  *psSuggestedString;
} DbvEnumeratedPrompt;
```

```
typedef struct DbvPromptId
{
DbvPromptIdType ePromptIdType;
union {
     DbvEnumeratedPrompt    *psEnumeratedPrompt;
     DbvInternationalString *psNonEnumeratedPrompt;
  } uPromptId;
} DbvPromptId;


typedef enum DbvPromptInfoType
{
    PIIT_Character = 1,
    PIIT_Encrypted = 2,
    PIIT_Invalid   = -1
} DbvPromptInfoType;



typedef struct DbvEncryption
{
DbvOctetString *psCryptType;
DbvOctetString *psCredential;
DbvOctetString  sData;
} DbvEncryption;



typedef struct DbvPromptInfo
{
DbvPromptInfoType ePromptInfoType;
union {
     DbvInternationalString *psCharacter;
     DbvEncryption          *psEncrypted;
  } uPromptInfo;
} DbvPromptInfo;



typedef enum DbvDataType {
   DTTT_Integer      = 1,
   DTTT_Date1        = 2,
   DTTT_Float        = 3,
   DTTT_AlphaNumeric = 4,
   DTTT_UrlUrn       = 5,
   DTTT_Boolean      = 6,
   DTTT_Invalid      = -1
} DbvDataType;


typedef struct DbvChallenge
{
DbvPromptId              sPromptId;
DbvInternationalString  *psDefaultResponse;
DbvPromptInfo           *psPromptInfo;
DbvInternationalString  *psRegExpr;
DbvNull                  pvResponseRequired; /* Set to either NULL or
                                   1 alloced byte */
```

```
DbvInternationalString **ppsAllowedValues;
DbvNull                  pvShouldSave; /* Set to either NULL or
                                1 alloced byte */
DbvDataType             *peDataType;
DbvASN1External         *psDiagnostic;
} DbvChallenge;


typedef enum DbvResponseType {
   RT_String      = 1,
   RT_Accept      = 2,
   RT_Acknowledge = 3,
   RT_Diagnostic  = 4,
   RT_Encrypted   = 5,
   RT_Invalid    = -1
} DbvResponseType;



/* NOTE: No union member for RT_Acknowledge */
typedef struct DbvPromptResponse
{
DbvResponseType eResponseType;
union {
     DbvInternationalString *psString;      /* RT_String */
     DbvBool                 fAccept;       /* RT_Accept */
     DbvDiagRec             *psDiagnostic;  /* RT_Diagnostic */
     DbvEncryption          *psEncryption;  /* RT_Encrypted */
  } uResponse;
} DbvPromptResponse;



typedef struct DbvResponse
{
DbvPromptId       sPromptId;
DbvPromptResponse sPromptResponse;
} DbvResponse;



typedef struct DbvPromptObject
{
DbvPromptObjectType ePromptObjectType;
union {
     DbvChallenge **ppsChallenge;
     DbvResponse  **ppsResponse;
   } uPromptObject;
} DbvPromptObject;
```

3.2.4 Character Set and Language negotiation Record Syntax

```
/* -------------------------------------------------------------------*/
/* ---- User Info Format "Character Set And Language Negotiation" ----- */

typedef enum DbvCharSetAndLanguageType {
   CSALT_Proposal = 1,
   CSALT_Response = 2,
   CSALT_Invalid  = -1
} DbvCharSetAndLanguageType;



typedef enum DbvProposedCharSetType {
   PRCST_Iso2022  = 1,
   PRCST_Iso10646 = 2,
   PRCST_Private  = 3,
   PRCST_Invalid  = -1
} DbvProposedCharSetType;



typedef enum DbvIso2022Type {
   I2022T_OriginProposal  = 1,
   I2022T_TargetResponse  = 2,
   I2022T_Invalid         = -1
} DbvIso2022Type;



typedef enum DbvEnvironmentType {
   ENT_SevenBit  = 1,
   ENT_EightBit = 2,
   ENT_Invalid  = -1
} DbvEnvironmentType;



typedef struct DbvEnvironment {
DbvEnvironmentType eEnvironmentType;
} DbvEnvironment;



typedef struct DbvInitialSet {
Ulong             ulG0;
Ulong             ulG1;
Ulong             ulG2;
Ulong             ulG3;
Ulong             ulC0;
Ulong             ulC1;
} DbvInitialSet;



typedef struct DbvLeftAndRight {
Ulong              ulGLeft;
Ulong              ulGRight;
} DbvLeftAndRight;
```

```
typedef struct DbvIso2022Proposal {
DbvEnvironment             *psProposedEnvironment;
Ulong                     **ppulProposedSets;
DbvInitialSet             **ppsProposedInitialSets;
DbvLeftAndRight            *psProposedLeftAndRight;
} DbvIso2022Proposal;



typedef struct DbvIso2022Response {
DbvEnvironment             *psSelectedEnvironment;
Ulong                     **ppulSelectedSets;
DbvInitialSet              *psSelectedInitialSet;
DbvLeftAndRight            *psSelectedLeftAndRight;
} DbvIso2022Response;



typedef struct DbvIso2022 {
DbvIso2022Type eIso2022Type;
union {
     DbvIso2022Proposal            *psOriginProposal;
     DbvIso2022Response            *psTargetResponse;
   } uIso2022Type;
} DbvIso2022;



typedef struct DbvIso10646 {
DbvObjectIdentifier            pcCollections;
DbvObjectIdentifier            pcEncodingLevel;
} DbvIso10646;



typedef enum DbvPrivateCharacterSetType {
   PCST_ViaOid                = 1,
   PCST_ExternallySpecified  = 2,
   PCST_PreviouslyAgreedUpon = 3,
   PCST_Invalid               = -1
} DbvPrivateCharacterSetType;




typedef struct DbvPrivateCharacterSet {
DbvPrivateCharacterSetType ePrivateCharacterSetType;
union {
     DbvObjectIdentifier            *ppcViaOid;
     DbvASN1External                *psExternallySpecified;
   } uPrivateCharacterSet;
} DbvPrivateCharacterSet;


typedef struct DbvProposedCharSet {
```

```
DbvProposedCharSetType eProposedCharSetType;
union {
     DbvIso2022             *psIso2022;
     DbvIso10646            *psIso10646;
     DbvPrivateCharacterSet *psPrivate;
   } uProposedCharSet;
} DbvProposedCharSet;



typedef struct DbvOriginProposal {
DbvProposedCharSet          **ppsProposedCharSets;
DbvLanguageCode             **ppsProposedLanguages;
DbvBool                      *pfRecordsInSelectedCharSets;
} DbvOriginProposal;



typedef enum DbvSelectedCharSetType {
   SCST_Iso2022  = 1,
   SCST_Iso10646 = 2,
   SCST_Private  = 3,
   SCST_None     = 4,
   SCST_Invalid  = -1
} DbvSelectedCharSetType;



typedef struct DbvSelectedCharSet {
DbvSelectedCharSetType eSelectedCharSetType;
union {
     DbvIso2022             *psIso2022;
     DbvIso10646            *psIso10646;
     DbvPrivateCharacterSet *psPrivate;
   } uSelectedCharSet;
} DbvSelectedCharSet;



typedef struct DbvTargetResponse {
DbvSelectedCharSet           *psSelectedCharSets;
DbvGeneralString             *psSelectedLanguage;
DbvBool                      *pfRecordsInSelectedCharSets;
} DbvTargetResponse;



typedef struct DbvCharSetAndLanguage {
DbvCharSetAndLanguageType eCharSetAndLanguageType;
union {
     DbvOriginProposal *psProposal;
     DbvTargetResponse *psResponse;
   } uCharSetAndLanguage;
} DbvCharSetAndLanguage;


#endif
```

4 Types and Sub-Structures in itemordr.h

```
/* ----------  itemordr.h -----------------------------------------*/

/* When using this file with other DBV-OSI II includes, include as follows

#include "sidefs.h"
#include "sitypes.h"
#include "sitypes2.h"
#include "dbvproto.h"
#include "iord_ext.h"
#include "itemordr.h"
*/
#ifdef PATCH
/*
#include "crxtypes.h"
*/
#endif /* divert(0) */




#if !defined(ITEMORDER_H)
#define ITEMORDER_H


typedef enum DbvEsIoAuxiliaryStatus {
             IOAS_NotReceived       = 1,
             IOAS_LoanQueue         = 2,
             IOAS_Forwarded         = 3,
             IOAS_UnfilledCopyright = 4,
             IOAS_FilledCopyright   = 5,
             IOAS_Invalid           = -1
} DbvEsIoAuxiliaryStatus;

typedef enum DbvEsIoPaymentMethodType {
             IOPM_BillInvoice          = 1,
             IOPM_Prepay               = 2,
             IOPM_DepositAccount       = 3,
             IOPM_CreditCard           = 4,
             IOPM_CardInfoPrevSupplied = 5,
             IOPM_PrivateKnown         = 6,
             IOPM_PrivateNotKnown      = 7,
             IOPM_Invalid              = -1
} DbvEsIoPaymentMethodType;


typedef struct DbvEsIoTargetPart {
DbvASN1External             *psItemRequest;        /* ILL ItemOrder PDU */
DbvASN1External             *psStatusOrErrorReport;
DbvEsIoAuxiliaryStatus      *peAuxiliaryStatus;
} DbvEsIoTargetPart;
```

```
typedef struct DbvEsIoResultSetItem {
DbvInternationalString    sResultSetId;
Ulong                     ulItem;
} DbvEsIoResultSetItem;


typedef struct DbvEsIoOriginPartNotToKeep {
DbvEsIoResultSetItem     *psResultSetItem;
DbvASN1External          *psItemRequest;     /* ILL ItemOrder */
} DbvEsIoOriginPartNotToKeep;


typedef struct DbvEsIoCreditCardInfo {
DbvInternationalString   sNameOnCard;
DbvInternationalString   sExpirationDate;
DbvInternationalString   sCardNumber;
} DbvEsIoCreditCardInfo;



typedef union uPaymentMethod {
     DbvEsIoCreditCardInfo  *psCreditCardInfo;
     DbvASN1External        *psPrivateNotKnown;
} uPaymentMethod;

typedef struct DbvEsIoPaymentMethod {
DbvEsIoPaymentMethodType          ePaymentMethodType;
uPaymentMethod                   *puPaymentMethod;
} DbvEsIoPaymentMethod;


typedef struct DbvEsIoAdditionalBilling {
DbvEsIoPaymentMethod         sPaymentMethod;
DbvInternationalString      *psCustomerReference;
DbvInternationalString      *psCustomerPONumber;
} DbvEsIoAdditionalBilling;


typedef struct DbvEsIoContact {
DbvInternationalString    *psName;
DbvInternationalString    *psPhone;
DbvInternationalString    *psEmail;
} DbvEsIoContact;


typedef struct DbvEsIoOriginPartToKeep {
DbvASN1External          *psSupplementalDescription;
DbvEsIoContact           *psContact;
DbvEsIoAdditionalBilling *psAdditionalBilling;
} DbvEsIoOriginPartToKeep;


typedef struct DbvEsIoRequest {
DbvEsIoOriginPartToKeep      *psToKeep;
DbvEsIoOriginPartNotToKeep     sNotToKeep;
} DbvEsIoRequest;
```

```
typedef struct DbvEsIoTaskPackage {
DbvEsIoOriginPartToKeep      *psOriginPart;
DbvEsIoTargetPart        sTargetPart;
} DbvEsIoTaskPackage;



typedef enum DbvItemOrderReqOrRsp {
                    IOROR_Request  = 1,
                    IOROR_Response = 2
} DbvItemOrderReqOrRsp;


typedef struct DbvItemOrderExtendedService {
DbvItemOrderReqOrRsp     eItemOrderReqRsp;
union {
    DbvEsIoRequest              *psItemOrderRequest;
    DbvEsIoTaskPackage          *psItemOrderResponse;
  }               uItemOrderReqOrRsp;
} DbvItemOrderExtendedService;
```

4.1 ILL definitions

```
/* -----------------   ILL-Definitions ------------------------- */

typedef enum  DbvIllStringType {
            ST_GeneralString = 1,
            ST_EdifactString = 2
} DbvIllStringType;



typedef struct DbvIllString {
DbvIllStringType  eStringType;
union {
      DbvGeneralString   *psGeneralString;
      DbvVisibleString    pcVisibleString;
    }                   uString;
} DbvIllString;




/* enumeration types IllRequest */

typedef char*  DbvPrintableString;

typedef enum DbvIllProtocolVersion {
            PV_Version1 = 1,
            PV_Version2 = 2,
            PV_Invalid  = -1
```

```
}DbvIllProtocolVersion;

typedef enum DbvNameOfPersOrInstType {
            NPI_NameOfPerson      = 1,
                NPI_NameOfInstitution = 2
} DbvNameOfPersOrInstType;


typedef enum  DbvPersOrInstSymbolType {
            SPI_PersonSymbol      = 1,
                SPI_InstitutionSymbol = 2
} DbvPersOrInstSymbolType;


typedef enum  DbvTransactionType {
            TT_Simple      = 1,
                TT_Chained     = 2,
                TT_Partitioned = 3,
                TT_Invalid     = -1
} DbvTransactionType;


typedef enum  DbvDeliveryServiceType {
            DST_PhysicalDelivery   = 1,
                DST_ElectronicDelivery = 2,
                DST_Invalid            = -1
} DbvDeliveryServiceType;


typedef enum DbvElDeliveryDetailsType {
            DDT_SystemAddress = 1,
            DDT_SystemId      = 2
} DbvElDeliveryDetailsType;


typedef enum  DbvServiceType {
            ST_Loan              = 1,
                ST_CopyNonReturnable = 2,
            ST_Locations         = 3,
            ST_Estimate          = 4,
            ST_ResponderSpecific = 5,
            ST_Invalid           = -1
} DbvServiceType;


typedef enum  DbvOptionalMsgRequirements {
            OMR_Requires = 1,
            OMR_Desires = 2,
            OMR_Neither = 3,
            OMR_Invalid = -1
} DbvOptionalMsgRequirements;


typedef enum  DbvExpiry {
            EXP_NeedBeforeDate = 1,
            EXP_OtherDate      = 2,
            EXP_NoExpiry       = 3,
            EXP_Invalid        = -1
```

```
} DbvExpiry;


typedef enum DbvSupplyMediumType {
            SMT_Printed              = 1,
            SMT_Photocopy            = 2,
            SMT_Microform            = 3,
            SMT_FilmOrVideoRecording = 4,
            SMT_AudioRecording       = 5,
            SMT_MachineReadable      = 6,
            SMT_Other                = 7,
            SMT_Invalid              = -1
} DbvSupplyMediumType;


typedef enum  DbvPlaceOnHold {
            POH_Yes              = 1,
            POH_No               = 2,
            POH_ResponderPolicy  = 3,
            POH_Invalid          = -1
} DbvPlaceOnHold;


typedef enum DbvItemType {
            IT_Monograph = 1,
            IT_Serial    = 2,
            IT_Other     = 3,
            IT_Invalid   = -1
} DbvItemType;


typedef enum  DbvMediumType {
            MT_Printed              = 1,
            MT_Microform            = 3,
            MT_FilmOrVideoRecording = 4,
            MT_AudioRecording       = 5,
            MT_MachineReadable      = 6,
            MT_Other                = 7,
            MT_Invalid              = -1
} DbvMediumType;


typedef enum  DbvPreference {
            PRE_Ordered   = 1,
            PRE_Unordered = 2,
            PRE_Invalid   = -1
} DbvPreference;


/* enumeration types StatusOrErrorReport */

typedef enum  DbvTransactionResults {
            TR_Conditional       = 1,
            TR_Retry             = 2,
            TR_Unfilled          = 3,
            TR_LocationsProvided = 4,
            TR_WillSupply        = 5,
            TR_HoldPlaced        = 6,
```

```
            TR_Estimate          = 7,
            TR_Invalid           = -1
} DbvTransactionResults;


typedef enum  DbvMostRecentService {
            MRS_IllRequest          = 1,
            MRS_Forward             = 21,
            MRS_ForwardNotification = 2,
            MRS_Shipped             = 3,
            MRS_IllAnswer           = 4,
            MRS_ConditionalReply    = 5,
            MRS_Cancel              = 6,
            MRS_CancelReply         = 7,
            MRS_Received            = 8,
            MRS_Recall              = 9,
            MRS_Returned            = 10,
            MRS_CheckedIn           = 11,
            MRS_Overdue             = 12,
            MRS_Renew               = 13,
            MRS_RenewAnswer         = 14,
            MRS_Lost                = 15,
            MRS_Damaged             = 16,
            MRS_Message             = 17,
            MRS_StatusQuery         = 18,
            MRS_StatusOrErrorReport = 19,
            MRS_Expired             = 20,
            MRS_Invalid             = -1
} DbvMostRecentService;


typedef enum  DbvShippedServiceType {
            SST_Loan              = 1,
            SST_CopyNonReturnable = 2,
            SST_Invalid           = -1
} DbvShippedServiceType;


typedef enum  DbvCurrentState {
            CS_NotSupplied        = 1,
            CS_Pending            = 2,
            CS_InProcess          = 3,
            CS_Forward            = 4,
            CS_Conditional        = 5,
            CS_CancelPending      = 6,
            CS_Cancelled          = 7,
            CS_Shipped            = 8,
            CS_Received           = 9,
            CS_RenewPending       = 10,
            CS_NotReceivedOverdue = 11,
            CS_RenewOverdue       = 12,
            CS_Overdue            = 13,
            CS_Returned           = 14,
            CS_CheckedIn          = 15,
            CS_Recall             = 16,
            CS_Lost               = 17,
            CS_Unknown            = 18,
            CS_Invalid            = -1
```

```
} DbvCurrentState;

typedef enum  DbvIllAPDUType {
            IAT_IllRequest           = 1,
            IAT_ForwardNotification = 2,
            IAT_Shipped              = 3,
            IAT_IllAnswer            = 4,
            IAT_ConditionalReply     = 5,
            IAT_Cancel               = 6,
            IAT_CancelReply          = 7,
            IAT_Received             = 8,
            IAT_Recall               = 9,
            IAT_Returned             = 10,
            IAT_CheckedIn            = 11,
            IAT_Overdue              = 12,
            IAT_Renew                = 13,
            IAT_RenewAnswer          = 14,
            IAT_Lost                 = 15,
            IAT_Damaged              = 16,
            IAT_Message              = 17,
            IAT_StatusQuery          = 18,
            IAT_StatusOrErrorReport = 19,
            IAT_Expired              = 20,
            IAT_Invalid              = -1
} DbvIllAPDUType;


typedef enum  DbvReportSource {
            RS_User     = 1,
            RS_Provider = 2
} DbvReportSource;


typedef enum  DbvIntermediaryProblem {
            IP_CannotSendOnward = 1,
            IP_Invalid          = -1
} DbvIntermediaryProblem;


typedef enum  DbvUnableToPerform {
            UTP_NotAvailable       = 1,
            UTP_ResourceLimitation = 2,
            UTP_Other              = 3,
            UTP_Invalid            = -1
} DbvUnableToPerform;


typedef enum  DbvUserErrorReportType {
            UER_AlreadyForwarded    = 1,
            UER_IntermediaryProblem = 2,
            UER_SecurityProblem     = 3,
            UER_UnableToPerform     = 4
} DbvUserErrorReportType;


typedef enum  DbvProviderErrorReportType {
            PER_GeneralProblem            = 1,
            PER_TransactionIdProblem      = 2,
```

```
            PER_StateTransitionProhibited = 3
} DbvProviderErrorReportType;


typedef enum  DbvTransactionIdProblem {
            TIP_DuplicateTransactionId = 1,
            TIP_InvalidTransactionId   = 2,
            TIP_UnknownTransactionId   = 3,
            TIP_Invalid                   = -1
} DbvTransactionIdProblem;

typedef enum DbvReasonNoReport {
            RNR_Temporary = 1,
            RNR_Permanent = 2,
            RNR_Invalid   = -1
} DbvReasonNoReport;

typedef enum  DbvGeneralProblem {
            GP_UnrecognizedAPDU              = 1,
            GP_MistypedAPDU                  = 2,
            GP_BadlyStructuredAPDU           = 3,
            GP_ProtocolVersionNotSupported = 4,
            GP_Other                         = 5,
            GP_Invalid                       = -1
} DbvGeneralProblem;

/* end enumeration types StatusOrErrorReport */
```

4.2 ILL Request

```
/* structures IllRequest */

/* SystemId  */

typedef struct DbvPersonOrInstitutionSymbol {
DbvPersOrInstSymbolType       eSymbolType;
DbvIllString                  sSymbol;
} DbvPersonOrInstitutionSymbol;


typedef struct  DbvNameOfPersonOrInstitution {
DbvNameOfPersOrInstType       eNameType;
DbvIllString                  sName;
} DbvNameOfPersonOrInstitution;


typedef struct  DbvSystemId {
DbvPersonOrInstitutionSymbol  *psSymbol;
DbvNameOfPersonOrInstitution  *psName;
} DbvSystemId;
```

```
/* TransactionId */

typedef struct  DbvTransactionId {
DbvSystemId             *psInitialRequesterId;
DbvIllString             sTransactionGroupQualifier;
DbvIllString             sTransactionQualifier;
DbvIllString            *psSubTransactionQualifier;
} DbvTransactionId;


/* ServiceDateTime */

typedef struct  DbvDateTime {
DbvVisibleString        pcDate;  /* YYYYMMDD,  mand. */
DbvVisibleString        pcTime;  /* HHMMSS, opt. */
} DbvDateTime;


typedef struct  DbvServiceDateTime {
DbvDateTime              sThisService;
DbvDateTime             *psOriginalService;
} DbvServiceDateTime;


/* DeliveryAddress, SystemAddress */

typedef struct  DbvPostalAddress {
DbvNameOfPersonOrInstitution  *psNameOfPersonOrInstitution;
DbvIllString                  *psExtendedPostalDeliveryAddress;
DbvIllString                  *psStreetAndNumber;
DbvIllString                  *psPostOfficeBox;
DbvIllString                  *psCity;
DbvIllString                  *psRegion;
DbvIllString                  *psCountry;
DbvIllString                  *psPostalCode;
} DbvPostalAddress;


typedef struct  DbvSystemAddress {
DbvIllString            *psTelecomServiceIdentifier;
DbvIllString            *psTelecomServiceAddress;
} DbvSystemAddress;


typedef struct  DbvDeliveryAddress {
DbvPostalAddress        *psPostalAddress;
DbvSystemAddress        *psElectronicAddress;
} DbvDeliveryAddress;


/* DeliveryService */

typedef struct  DbvElDeliveryDetails {
DbvElDeliveryDetailsType      eDeliveryDetailsType;
union {
      DbvSystemAddress            *psSystemAddress;
      DbvSystemId                 *psSystemId;
```

```
    }                   uDeliveryDetails;
} DbvElDeliveryDetails;



typedef union uDeliveryParameter {
      DbvASN1External        *psDummy;  /* mand. */
} uDeliveryParameter;


typedef struct  DbvElDeliveryParameter {
DbvObjectIdentifier      pcDeliveryMode;       /* opt. */
uDeliveryParameter      *puDeliveryParameter;  /* opt. */
} DbvElDeliveryParameter;


typedef union uDocumentTypeParameter {
      DbvASN1External        *psDummy;  /* mand. */
} uDocumentTypeParameter;


typedef struct  DbvDocumentTypeParameter {
DbvObjectIdentifier      pcDocumentTypeId;        /* opt. */
uDocumentTypeParameter  *puDocumentTypeParameter; /* opt. */
} DbvDocumentTypeParameter;


typedef struct  DbvElectronicDeliveryService {
DbvElDeliveryParameter       *psDeliveryParameter;
DbvDocumentTypeParameter     *psDocumentTypeParameter;
DbvIllString                 *psDeliveryDescription;
DbvElDeliveryDetails          sDeliveryDetails;
DbvIllString                 *psNameOrCode;
DbvVisibleString         pcDeliveryTime;    /* opt. */
} DbvElectronicDeliveryService;


typedef struct  DbvDeliveryService {
DbvDeliveryServiceType       eDeliveryServiceType;
union {
      DbvIllString                  *psPhysicalDelivery;
      DbvElectronicDeliveryService  **ppsElectronicDelivery;
    }                   uDeliveryService;
} DbvDeliveryService;


/* RequesterOptionalMessageType */

typedef struct  DbvRequesterOptionalMessages {
DbvBool                      fCanSendReceived;
DbvBool                      fCanSendReturned;
DbvOptionalMsgRequirements   eRequesterShipped;
DbvOptionalMsgRequirements   eRequesterCheckedIn;
} DbvRequesterOptionalMessages;


typedef struct  DbvSearchType {
DbvIllString            *psLevelOfService;
```

```
DbvVisibleString   pcNeedBeforeDate;  /* opt. */
DbvExpiry         *peExpiryFlag;      /* default 3 */
DbvVisibleString   pcExpiryDate;      /* opt. */
} DbvSearchType;


/* SupplyMediumInfoType */

typedef struct  DbvSupplyMediumInfo {
DbvSupplyMediumType      eSupplyMediumType;
DbvIllString            *psMediumCharacteristics;
} DbvSupplyMediumInfo;


/* ClientId */

typedef struct  DbvClientId {
DbvIllString            *psClientName;
DbvIllString            *psClientStatus;
DbvIllString            *psClientIdentifier;
} DbvClientId;


/* ItemId */

typedef struct  DbvItemId {
DbvItemType       *peItemType;
DbvMediumType           *peMediumType;
DbvIllString            *psCallNumber;
DbvIllString            *psAuthor;
DbvIllString            *psTitle;
DbvIllString            *psSubTitle;
DbvIllString            *psSponsoringBody;
DbvIllString            *psPlaceOfPublication;
DbvIllString            *psPublisher;
DbvIllString            *psSeriesTitleNumber;
DbvIllString            *psVolumeIssue;
DbvIllString            *psEdition;
DbvIllString            *psPublicationDate;
DbvIllString            *psPublicationDateOfComponent;
DbvIllString            *psAuthorOfArticle;
DbvIllString            *psTitleOfArticle;
DbvIllString            *psPagination;
DbvASN1External         *psNationalBibliographyNo;
DbvIllString            *psIsbn;
DbvIllString            *psIssn;
DbvASN1External         *psSystemNo;
DbvIllString            *psAdditionalNoLetters;
DbvIllString            *psVerificationReferenceSource;
} DbvItemId;


/* CostInfoType */

typedef struct DbvAmount {
DbvPrintableString      pcCurrencyCode;    /* opt. */
DbvPrintableString      pcMonetaryValue;   /* mand */
} DbvAmount;
```

```
typedef struct DbvCostInfo {
DbvIllString            *psAccountNumber;
DbvAmount               *psMaximumCost;
DbvBool                 *pfReciprocalAgreement;  /* default FALSE */
DbvBool                 *pfWillPayFee;           /* default FALSE */
DbvBool                 *pfPaymentProvided;      /* default FALSE */
} DbvCostInfo;


/* ThirdPartyInfoType */

typedef struct DbvSendToList {
DbvSystemId              sSystemId;
DbvIllString            *psAccountNumber;
DbvSystemAddress  *psSystemAddress;
} DbvSendToList;


typedef struct DbvThirdPartyInfo {
DbvBool                  *pfPermissionToForward;          /* default FALSE */
DbvBool                  *pfPermissionToChain;            /* default FALSE */
DbvBool                  *pfPermissionToPartition;        /* default FALSE */
DbvBool                  *pfPermissionToChangeSendToList; /* default FALSE */
DbvSystemAddress         *psInitialRequesterAddress;
DbvPreference            *pePreference;                   /* default 2 */
DbvSendToList           **ppsSendToList;
DbvSystemId             **ppsAlreadyTriedList;
} DbvThirdPartyInfo;

typedef struct DbvExtension {
Ulong                    ulIdentifier;
DbvBool                 *pfCritical;     /* default FALSE */
union {
      DbvSubitoAdditionalInfo *psAddIllReqInfo;
    }            uItem;
} DbvExtension;


typedef struct DbvIllRequest {
DbvIllProtocolVersion          eProtocolVersion;
DbvTransactionId               sTransactionId;
DbvServiceDateTime             sServiceDateTime;
DbvSystemId                   *psRequesterId;
DbvSystemId                   *psResponderId;
DbvTransactionType            *peTransactionType;     /* default 1 */
DbvDeliveryAddress            *psDeliveryAddress;
DbvDeliveryService            *psDeliveryService;
DbvDeliveryAddress            *psBillingAddress;
DbvServiceType               **ppeServiceType;
DbvASN1External               *psResponderSpecificService;
DbvRequesterOptionalMessages   sRequesterOptionalMessages;
DbvSearchType                 *psSearchType;
DbvSupplyMediumInfo          **ppsSupplyMediumInfo;
DbvPlaceOnHold                *pePlaceOnHold;       /* default 3 */
DbvClientId                   *psClientId;
DbvItemId                      sItemId;
```

```
DbvASN1External             **ppsSupplementalItemDescription;
DbvCostInfo                  *psCostInfo;
DbvIllString                 *psCopyrightCompliance;
DbvThirdPartyInfo            *psThirdPartyInfo;
DbvBool                      *pfRetry;  /* default FALSE */
DbvBool                      *pfForward;      /* default FALSE */
DbvIllString                 *psRequesterNote;
DbvIllString                 *psForwardNote;
DbvExtension                **ppsRequestExtension;
} DbvIllRequest;
```

4.3 ILL Status or Error Report

```
/* structures StatusOrErrorReport */

typedef struct  DbvHistoryReport {
DbvVisibleString             pcDateRequested;           /* opt. */
DbvIllString                *psAuthor;
DbvIllString                *psTitle;
DbvIllString                *psAuthorOfArticle;
DbvIllString                *psTitleOfArticle;
DbvVisibleString             pcDateOfLastTransition;    /* opt. */
DbvMostRecentService         eMostRecentService;
DbvVisibleString             pcDateOfMostRecentService; /* mand. */
DbvSystemId                  sSystemId;
DbvShippedServiceType       *peShippedServiceType;
DbvTransactionResults       *peTransactionResults;
DbvIllString                *psMostRecentServiceNote;
} DbvHistoryReport;


typedef struct  DbvStateTransitionProhibited {
DbvIllAPDUType              eAPDUType;
DbvCurrentState             eCurrentState;
} DbvStateTransitionProhibited;


typedef struct  DbvAlreadyForwarded {
DbvSystemId                  sResponderId;
DbvSystemAddress            *psResponderAddress;
} DbvAlreadyForwarded;


typedef struct  DbvProviderErrorReport {
DbvProviderErrorReportType    eReportType;
union {
      DbvGeneralProblem             eGeneralProblem;
      DbvTransactionIdProblem       eTransactionIdProblem;
      DbvStateTransitionProhibited *psStateTransitionProblem;
    }                   uProviderErrorReport;
} DbvProviderErrorReport;
```

```
typedef struct  DbvUserErrorReport {
DbvUserErrorReportType             eReportType;
union {
      DbvAlreadyForwarded              *psAlreadyForwarded;
      DbvIntermediaryProblem            eIntermediaryProblem;
      DbvIllString                     *psSecurityProblem;
      DbvUnableToPerform                eUnableToPerform;
    }                               uUserErrorReport;
} DbvUserErrorReport;

typedef struct DbvUserOrProviderErrorReport {
DbvReportSource              eReportSource;
union {
      DbvProviderErrorReport       *psProviderErrorReport;
      DbvUserErrorReport           *psUserErrorReport;
    }             uReport;
} DbvUserOrProviderErrorReport;

typedef struct  DbvErrorReport {
DbvIllString                  sCorrelationInfo;
DbvUserOrProviderErrorReport  sReport;
} DbvErrorReport;


typedef struct  DbvStatusReport {
DbvHistoryReport             sUserStatusReport;
DbvCurrentState              eProviderStatusReport;
} DbvStatusReport;


typedef struct  DbvStatusOrErrorReport {
DbvIllProtocolVersion          eProtocolVersion;
DbvTransactionId               sTransactionId;
DbvServiceDateTime             sServiceDateTime;
DbvSystemId                   *psRequesterId;
DbvSystemId                   *psResponderId;
DbvReasonNoReport             *peReasonNoReport;
DbvStatusReport               *psStatusReport;
DbvErrorReport                *psErrorReport;
DbvIllString                  *psNote;
DbvExtension                 **ppsStOrErReportExtension;
} DbvStatusOrErrorReport;
```

4.4 ILL APDU Definition Structure

```
/************************************************/
/* ILL APDU Definition */

/* Note ILL_Invalid in the typedef enum below.
   An Invalid option is not normally
   used in DBV-OSI II union enums but is needed here because
```

```
   DbvIllReport only supports two of the many possible
   ILL Report types which may be received.

   ILL_Invalid should never be used when setting up
   an ILL request for sending, it will be set
   internally by the API if an ILL report unsupported
   by DBV-OSI II is received.

*/

typedef enum DbvIllPduType {
             ILL_IllRequest              = 1,
             ILL_StatusOrErrorReport     = 19,
             ILL_Invalid                 = -1
} DbvIllPduType;



typedef struct DbvIllPdu {
DbvIllPduType          eIllPduType;
union {
    struct DbvIllRequest               *psIllRequest;
    struct DbvStatusOrErrorReport      *psStatusOrErrorReport;
  }              uIllPdu;
} DbvIllPdu;



#endif
```

5 Types and Sub-Structures in iord_ext.h

```
/* -----------  iord_ext.h --------------------------------------------*/

/* When using this file with other DBV-OSI II includes, include as follows

#include "sidefs.h"
#include "sitypes.h"
#include "sitypes2.h"
#include "dbvproto.h"
#include "iord_ext.h"
#include "itemordr.h"
*/
#ifdef PATCH
/*
#include "crxtypes.h"
*/
#endif /* divert(0) */




/*  Module Name: @(#)iord_ext.h     7.2
    Version: 12/2/96
    Author:

    Crossnet Stystems Ltd.
    Unit 3, The Paddock,
    Hambridge Road,
    Newbury,
    Berkshire, RG14 5TQ

    Date of Issue:

    QA Acceptance:
    Date of Acceptance:

    Description:


*/

#if !defined(IORD_EXT_H)
#define IORD_EXT_H
```

5.1 SUBITO Item order Extensions

```
/* ---------------------------------------------------------------------
```

```
            C-Definitions for Subito ItemOrder Extensions
----------------------------------------------------------------------- */

typedef enum DbvRequesterGroup {
            RG_UserGroup1            = 1,
            RG_UserGroup2            = 2
} DbvRequesterGroup;


typedef enum DbvStatusLevelUser {
            SLU_All           = 1,
            SLU_Negative      = 2,
            SLU_None          = 3
} DbvStatusLevelUser;


typedef enum DbvStatusLevelRequester {
            SLO_All           = 1,
            SLO_Negative      = 2,
            SLO_None          = 3
} DbvStatusLevelRequester;


typedef enum DbvBillingInvoiceType {
            BIT_Single        = 1,
            BIT_Collective    = 2
} DbvBillingInvoiceType;


typedef enum DbvAddBillingType {
            ABT_BankInfo                  = 1,
            ABT_CreditCardInfo            = 2,
            ABT_InvoiceType               = 3,
            ABT_PrepaidType               = 4,
            ABT_DepositType               = 5
} DbvAddBillingType;



/* ------------------------------------------------------------------
   additional requester information
---------------------------------------------------------------------*/

typedef struct DbvAddRequesterInfo {
DbvRequesterGroup *peRequesterGroup;          /* default RG_UserGroup1 */
DbvVisibleString   pcRequesterEmailAddress;   /* opt. */
} DbvAddRequesterInfo;


/* ------------------------------------------------------------------
   additional delivery information
---------------------------------------------------------------------*/


typedef struct DbvAddDeliveryInfo {
DbvStatusLevelUser       *peStatusLevelUser;         /* default SLU_Negative */
DbvStatusLevelRequester  *peStatusLevelOrigin;       /* default SLO_None */
DbvVisibleString    pcDeliveryNotes;            /* opt. */
```

```
DbvVisibleString    pcCountryDeliveryTarget;    /* opt */
DbvBool                  *pfAlternativePostalDelivery;/* default FALSE */
} DbvAddDeliveryInfo;


/* ---------------------------------------------------------------------
   additional billing information
---------------------------------------------------------------------*/


typedef struct DbvBillingBankInfo {
DbvVisibleString         pcBankAccountHouse;          /* mand */
DbvVisibleString         pcBankAccountRoutingSymbol;  /* mand */
DbvVisibleString         pcBankAccountNo;             /* mand */
DbvBool                  fBankAuthority;
} DbvBillingBankInfo;




typedef struct DbvBillingCreditCardInfo {
DbvVisibleString  pcCardCompany;          /* mand */
DbvVisibleString  pcCardNo;               /* mand */
DbvVisibleString  pcCardValidFrom;        /* opt */
DbvVisibleString  pcCardValidThru;        /* mand */
} DbvBillingCreditCardInfo;


typedef struct DbvAddBillingInfo {
DbvAddBillingType        eAddBillingType;
union {
      DbvBillingBankInfo            *psBillingBankInfo;
      DbvBillingCreditCardInfo      *psBillingCreditCardInfo;
      DbvBillingInvoiceType          eBillingInvoiceType;
      DbvVisibleString               pcPrepaidId;
      DbvVisibleString               pcDepositId;
  }                     uAddBillingInfo;
} DbvAddBillingInfo;


typedef struct DbvSubitoAdditionalInfo {
DbvBool                       *pfCustomerInfoToKeep; /* default FALSE */
DbvAddRequesterInfo           *psAddRequesterInfo;
DbvAddDeliveryInfo            *psAddDeliveryInfo;
DbvAddBillingInfo       *psAddBillingInfo;
} DbvSubitoAdditionalInfo;



#endif
```