

ZIG Tutorial
Copenhagen
August 1997

Z39.50
Extended Services

Ray Denenberg
Library of Congress
ray@rden.loc.gov

Extended services

- **Save a result set**
- **Export**
- **Periodic Query**
- **Order an item**
- **Update a database**

Extended Services Facility

- the ES *service*
- the *extended services*
- The ES *database*
(*Task Packages*)

The Extended Services *Service* (the ES operation)

Origin

Target

Extended Service

request >

< *Extended Services
Response*

Analogous to a Search operation:

Search request >

< **Search response**

Origin

Target

Extended Service request >

< *Access Control Request*

***Access Control response* >**

< **Extended Services response**

*In the ES database,
a "task Package" includes:*

- **Package type**
 - **persistent result set**
 - **Item Order**
 - **periodic query, etc.**
- **Package name**
- **User Id**
- **Retention time**

Task Package (continued)

- **Permissions**
- **Description**
- **Creation date/time**
- **Task status**
 - **pending**
 - **active**
 - **completed**
 - **aborted**
- **"Task Specific" Parameters**

Attribute Set Ext-1

<u>Attribute Type</u>	<u>Value</u>
Use	1
Permissions	2

Permission attribute used only when the value of the Use attribute is UserId, to search for task packages for which the specified user has the specified permission.

<u>Use</u>	<u>Value</u>	<u>Use</u>	<u>Value</u>
UserId	1	PackageType	5
PackageName	2	Reten. Time	6
CreationDate	3	TargetRef.	7
TaskStatus	4		

Extended Service request >
(< . . Extended Services response)

Search request >
< Search response

Present request >
< Present response

Example

Search *Es database for records
where 'packageType' =
'itemOrder' AND
'taskStatus' = 'pending*

Present *Records 1 through n
using ES record syntax.*

Origin

Target

Extended Service request >

Function

Type

Name

User

Retention

Permissions

Description

Task-specific-parameters

Wait action

Elements

Target Response

**Origin< . . . Target >Task
Package**

- **Operation Status:**
 - **Done**
 - **Accepted**
 - **Failure**
- **Operation Diagnostics**
- **(Task Package?)**
- **Task Status:**
 - **pending**
 - **active**
 - **complete**
 - **aborted**
- **P a c k a g e Diagnostics**
- **Reference**
- **C r e a t i o n date/time**

Wait-action

Wait Target *must* perform task before issuing ES response, and *include* Task-package in response.

wait if possible Target *requested to* perform task before issuing ES response (and *include* the task package). If target unwilling, proceed as though is were 'do not wait'.

don't wait Target *not* requested to perform task before issuing ES response (and include task package). But if it does, ok.

don't send pkg Target may perform task when it chooses, but *don't include task package*.

Z39-50

OBJECT IDENTIFIER

::= {1 2 840 10003}

Z39-50-extendedServices

OBJECT IDENTIFIER

::= {Z39-50 9}

PersistentResultSet	1
PersistentQuery	2
PeriodicQuerySchedule	3
ItemOrder	4
DatabaseUpdate	5
ExportSpecification	6
ExportInvocation	7

Example

**OID for ItemOrder ES definition:
1.2.840.10003.9.4**

Locally defined Z39.50 Extended Services

{1 2 840 10003 9 1000 m n)

m = an implementor id

**n = a number assigned by
that implementor**

e.g.

Extended service: Adloan

**Object identifier: 1.2.840.10003.9.
1000.108.1**

Implementor: Databasix

**Description: Library loans
system running
on Adserver**

The OID for an extended service:

- (1) • Identifies the ES type
- Identifies the ES definition,
Which defines:
 - (2) ■ task-type-specific
information occurring
in the ES request
 - (3) ■ task-type-specific
information occurring
in the task package

e.g.

```
ItemOrder ::= CHOICE{  
  esRequest      [1] IMPLICIT SEQUENCE{ .....  
  taskPackage    [2] IMPLICIT SEQUENCE{ .....  
}
```

(See page 60-61)

ExtendedServicesRequest ::= SEQUENCE{

referenceId

**function [3] IMPLICIT INTEGER {
 create (1),
 delete (2),
 modify (3)},**

pkgType [4] IMPLICIT OBJECT IDENTIFIER,

**pkgName [5] IMPLICIT InternationalString optional,
userId [6] IMPLICIT InternationalString optional,
retention [7] IMPLICIT IntUnit OPTIONAL,
prmissions [8] IMPLICIT Permissions OPTIONAL,
description [9] IMPLICIT InternationalString optional,**

taskSpecificParameters [10] EXTERNAL

**-- For the 'EXTERNAL', use OID of
-- specific ES definition and select CHOICE [1]:
-- 'esRequest'.**

waitAction [11]

.....

ExtendedServicesResponse ::= SEQUENCE{

referenceId

operation-status [3] IMPLICIT INTEGER{
done (1),
accepted (2),
failure (3)},

diagnostics [4]

taskPackage [5] EXTERNAL

optional,

-- Use OID: {Z39-50-recordSyntax (106)}
-- and corresponding syntax. For the
-- EXTERNAL, 'taskSpecific', within that
-- definition, use OID of the specific es, and
-- choose [2], 'taskPackage'.

otherInfo
}

RecordSyntax-ESTaskPackage
{Z39-50-recordSyntax ESTaskPackage (106)}

DEFINITIONS ::=

TaskPackage ::= SEQUENCE{

pkgType [1] IMPLICIT OBJECT IDENTIFIER,

-- oid of specific ES definition

pkgName [2] IMPLICIT InternationalString optional,

userId [3] IMPLICIT InternationalString optional,

retention [4] IMPLICIT IntUnit optional,

prmissions [5] IMPLICIT Permissions optional,

dscription [6] IMPLICIT InternationalString optional,

targetRef [7] IMPLICIT OCTET STRING optional,

creation [8] IMPLICIT GeneralizedTime optional,

taskStatus [9] IMPLICIT INTEGER{

pending (0),

active (1),

complete (2),

aborted (3)},

packageDiagnostics [10]

taskSpecificParameters [11] EXTERNAL

-- Use oid for specific ES definition (same oid as

-- packageType above) and select [2]

"taskPackage."

}

Specific Parameters for Item Order

Specific Task <u>Parameter</u> requested Item	<u>Origin Supplied</u>	<u>Target Supplied</u>	<u>Task Package Parameter</u>
	X		
itemRequest		X	X
supplemental Description	X		X
contact Information	X		X
additional BillingInfo	X		X
statusOr ErrorReport		X	X
auxiliaryStatus		X	X

OriginPartToKeep ::= SEQUENCE{

supplDescription [1] IMPLICIT EXTERNAL OPTIONAL,

contact [2] IMPLICIT SEQUENCE{

name [1] ...

phone [2] ...

email [3]

addlBilling [3] IMPLICIT SEQUENCE{

paymentMethod [1] CHOICE{

billInvoice IMPLICIT NULL,

prepay IMPLICIT NULL,

depositAccount IMPLICIT NULL,

CreditCardInfo,

cardInfoPreviouslySupplied

privateKnown IMPLICIT NULL,

privateNotKnown IMPLICIT EXTERNAL},

customerReference [2] } OPTIONAL}

CreditCardInfo ::= SEQUENCE{

nameOnCard [1]

expirationDate [2]

cardNumber [3]

OriginPartNotToKeep ::= SEQUENCE{

```
resultSetItem [1] IMPLICIT SEQUENCE{
    resultSetId [1] IMPLICIT InternationalString,
    item        [2] IMPLICIT INTEGER}
    OPTIONAL,
itemRequest [2] IMPLICIT EXTERNAL OPTIONAL
    -- When itemRequest is an ILL-Request APDU,
    -- use OID {iso standard 10161 abstract-syntax
    -- (2) ill-apdus (1)}
}
```

TargetPart ::= SEQUENCE{

```
itemRequest [1] IMPLICIT OPTIONAL,
    -- When itemRequest is an ILL-Request APDU,
    -- use OID 1.0.10161.2.1 (as above)
statusOrErrorReport [2] IMPLICIT EXTERNAL OPTIONAL,
auxiliaryStatus [3] IMPLICIT INTEGER{
    notReceived (1),
    loanQueue (2),
    forwarded (3),
    unfilledCopyright (4),
    filledCopyright (5)}
    OPTIONAL}
```

Specific Parameters for Database Update

Specific Task Parameter	Origin Supplied	Target Supplied	Task Package Parameter
action	X		X
database Name	X		X
schema	X		X
supplied Records	X		
record Ids	X		
supple- mental Ids	X		

Specific Parameters for Database Update (Continued)

<u>Specific Task Parameter</u>	<u>Origin Supplied</u>	<u>Target Supplied</u>	<u>Task Package Parameter</u>
correlation Info	X		X
elementSetName	X		X
updateStatus		X	X
global Diagnostics		X	X
taskPackage Records		X	X
record Statuses		X	X

OriginPartToKeep ::= SEQUENCE{

action [1] IMPLICIT INTEGER{
 recordInsert (1),
 recordReplace (2),
 recordDelete (3),
 elementUpdate (4)},

databaseName [2]

schema [3]
 OBJECT IDENTIFIER

elementSetName [4]

OriginPartNotToKeep

::=

SuppliedRecords

SuppliedRecords ::=

SEQUENCE OF SEQUENCE{

recordId [1] CHOICE{

- number** [1] ...
- string** [2] ...
- opaque** [3] ...

supplementalId [2] CHOICE{

- timeStamp** [1] ...
- versionNumber** [2] ...
- previousVersion** [3] ...

correlationInfo [3]

IMPLICIT CorrelationInfo ...

record [4]

IMPLICIT EXTERNAL}

CorrelationInfo ::= SEQUENCE{

-- origin may supply one or both for any record:

- note** [1]
- id** [2]

TargetPart ::= SEQUENCE{

updateStatus [1]
 success (1),
 partial (2),
 failure (3)},

globalDiagnostics [2]

-- These are non-surrogate diagnostics relating
-- to the task, not to individual records.

taskPackageRecords [3]

**IMPLICIT SEQUENCE OF
TaskPackageRecordStructure**

-- There should be a TaskPackageRecordStructure for
-- every record supplied. The target should create such
-- a structure for every record immediately upon
-- creating the task package to include correlation
-- information and status. The record itself would not
-- be included until processing for that record is
-- complete.

}

TaskPackageRecordStructure

::= SEQUENCE{

recordOrSurDiag [1] CHOICE {

record [1] IMPLICIT EXTERNAL,

-- Choose 'record' if recordStatus is 'success',

-- and elementSetName was supplied.

diagnostic [2] DiagRec

-- Choose 'diagnostic', if RecordStatus is failure.

} OPTIONAL,

-- The parameter recordOrSurDiag will thus be omitted only if

-- 'elementSetName' was omitted and recordStatus is 'success'; or if

-- record status is 'queued' or in 'process'.

correlationInfo [2] ... CorrelationInfo ...

-- This should be included if it was supplied by the origin.

recordStatus [3] IMPLICIT INTEGER{

success (1),

queued (2),

inProcess (3),

failure (4)}}}