

Metadata Replication Service

Technical Specifications

Version 2.2

**Data Centre and Web Services Division
National Informatics Centre
Department of Information Technology**

Table Contents

1. INTRODUCTION.....	3
1.1. SUMMARY OF REQUIREMENTS.....	3
1.1.1. <i>Metadata Replication Service (MDR Service)</i>	3
1.1.2. <i>Metadata Consolidation Server (MDC Server)</i>	4
2. METADATA CONSOLIDATION ARCHITECTURE.....	5
2.1. STANDARD METADATA SCHEMA	6
2.2. MDR SERVICE SPECIFICATIONS	8
2.2.1. <i>getStatus</i>	12
2.2.2. <i>getWebPageMetadata</i>	13
2.2.3. <i>getMetadataSize</i>	13
2.2.4. <i>getAllMetadata</i>	13
2.2.5. <i>getNewMetadata</i>	14
2.2.6. <i>getUpdatedMetadata</i>	14
2.2.7. <i>getExitedWebpages</i>	14
2.2.8. <i>notifyWebsiteAdmin</i>	14
2.3. MESSAGE LEVEL SECURITY.....	15
2.4. MDC-CMS-API SPECIFICATIONS	16
2.4.1. <i>connect</i>	17
2.4.2. <i>disconnect</i>	17
2.4.3. <i>startTransaction</i>	17
2.4.4. <i>commitTransaction</i>	18
2.4.5. <i>abortTransaction</i>	18
2.4.6. <i>getWebpageMetadata</i>	18
2.4.7. <i>getAllMetadata</i>	18
2.4.8. <i>getNewMetadata</i>	18
2.4.9. <i>getUpdatedMetadata</i>	20
2.4.10. <i>getExitedWebpages</i>	20
3. MDR SERVICE	21
3.1. MDRSERVICE.XML	21
3.2. METADATA.XML	22
4. MDC SERVER.....	24
4.1. MDR SERVICE STATUS MONITOR	24
4.1.1. <i>MdcServer.xml</i>	24
4.2. METADATA CONSOLIDATOR.....	26
5. MDC-CMS-API IMPLEMENTATION.....	26
6. CHECKLIST FOR MDC SERVICE.....	27

1. Introduction

National Portal of India or India Portal provides a single window access to information and services being provided by the various Indian Government entities. It is a Mission Mode Project under the National E-Governance Plan. It is designed and maintained by [National Informatics Centre \(NIC\)](#), DIT, MoCIT, Government of India. It provides consolidated view of content from various Government Ministries and Departments, at the Central/State/District level.

National Portal will enable integration of content from various government portals so that one-stop source for all information and services pertaining to various Government departments can be provided to citizens. For this purpose metadata from various government websites need to be consolidated on a periodic basis. This consolidated metadata will enable content consolidation for National portal and richer search experience for the user of National Portal.

1.1. Summary of Requirements

1.1.1. Metadata Replication Service (MDR Service)

MDR service would be an integral part of State Portal. It is a must for:

- Realizing metadata based content integration
- Consolidating metadata from all State Portals
- Integrating State Portals with National portal and
- Providing metadata based content discovery functionality

Primary objective of MDR service is to replicate metadata from State Portal to consolidated metadata repository. It would implement a well defined interface providing standard set of functionality as mentioned below. Its implementation may vary from State Portal to State Portal depending on the specific content management and technologies used for implementing State Portal.

MDR service must provide following functionality

- a. It should be implemented as a web service
- b. It should implement a well defined WSDL compliant standard interface.
- c. Provide metadata of published content from the content repository of State Portal, for following publishing events occurring in the given time period
 - i. New web page is published
 - ii. Metadata of published web pages is modified
 - iii. Web page is exited or deleted
- d. Normalize metadata extracted from content repository, based on defined standard metadata schema (this include attribute names, data type, number of attributes, valid values, max allowed length, data range etc.).
- e. Provide metadata of given web pages from the content repository of State Portal.
- f. Provide status of MDR service, indicating it's availability and normal functioning.
- g. Provide information about metadata validation errors (if any).
- h. Maintain time upto which metadata is propagated to "consolidated metadata repository".
- i. It should be deployed on a highly available infrastructure.
- j. It should provide message level authentication based on userid and password.
- k. Ensure that metadata is kept for sufficient time such that request for metadata can be satisfied as and when they come.
- l. Provide required error logging, activity logging and email notifications for errors.

MDR service should be implemented such that code specific to content repository, content management system and implementation of content publishing workflow is well encapsulated, so that migration from one content management system to another content management system should have

- No impact on "metadata consolidation server" and
- Minimal impact on "metadata replication service"

1.1.2. Metadata Consolidation Server (MDC Server)

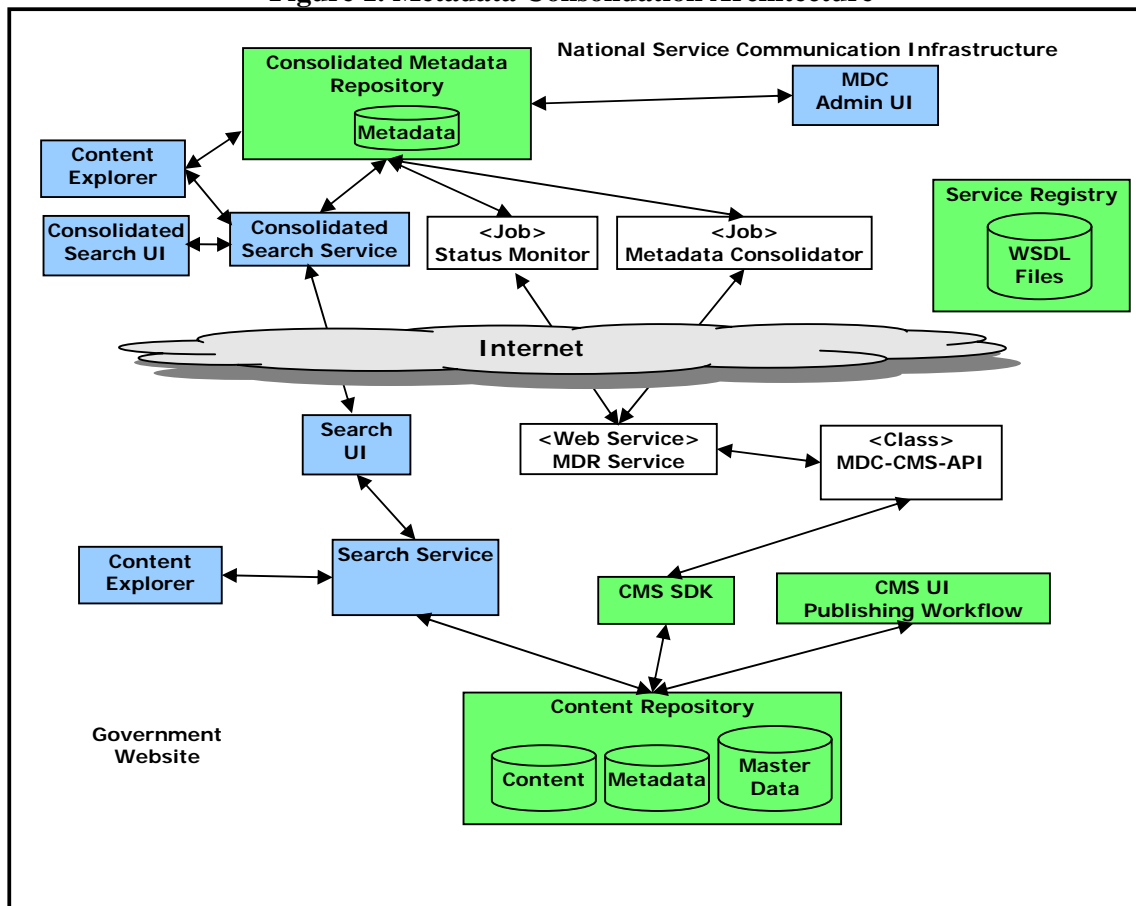
Metadata consolidation server would be a scheduled job type of entity. It would communicate with MDR services using a well defined standard interface.

MDC server would provide following functionality:

- a. Retrieve metadata of published content on State Portal's using MDR services at configurable periodicity.
- b. Metadata should be retrieved for following publishing events
 - i. New web page is published
 - ii. Metadata of published web pages is modified
 - iii. Web page is exited or deleted
- c. Validate retrieved metadata
- d. Normalize metadata based on defined standard metadata schema (this include attribute names, data type, number of attributes, valid values, max allowed length, data range etc.).
- e. Store retrieved metadata into consolidated metadata repository.
- f. Monitor the status of MDR services at configurable periodicity
- g. Maintain list of MDR services or State Portals from which metadata to be consolidated
- h. Maintain time upto which metadata is received from each MDR service
- i. Provide alerts for various errors such as
 - i. Non conformance to defined standard of metadata schema
 - ii. Non conformance to defined standard of master data
 - iii. Failing to normalize received metadata
 - iv. Failing to save received metadata to "consolidated metadata repository"
 - v. Failing to invoke MDR services
- j. Provide means to take one or more of following actions on detection of above errors
 - i. Log errors to defined log file or database
 - ii. Display message on system console or event log
 - iii. Send email message to defined email IDs (such as web master of State Portals)
 - iv. Send escalation email message if error condition persists beyond defined amount of time.
 - v. Recovery from errors
- k. Ensure that metadata is retrieved even when
 - i. MDR service temporarily goes down or not reachable due to reasons such as (including but not limited to)
 1. Internet connection failure
 2. Hardware failure
 3. Operating system failure
 4. Web server failure
 5. Application server failure
 6. Content repository failure
 7. Web service software failure etc.
 - ii. MDC server goes down temporarily or fails to invoke "MDR service"

2. Metadata Consolidation Architecture

Figure 1. Metadata Consolidation Architecture



Metadata consolidation would be achieved using a Metadata Replication Service (MDR Service), deployed on each of the website, from which metadata is required to be consolidated. MDR service would be developed as a web service. It would implement a well defined WSDL interface.

MDR Service communicates with content repository using MDC-CMS-API. This would implement a well defined interface called MDC-CMS-API. It encapsulates all communication with content repository. It would be responsible for retrieving metadata from content repository and converting it into a normalized form, which MDR service can send it to MDC server.

MDC server consist two jobs namely "Status Monitor" and "Metadata Consolidator". Status monitor would be responsible for monitoring the status of MDR services. Metadata consolidator would be responsible for retrieving metadata from MDR services.

2.1. Standard Metadata Schema

Following table describes the metadata attributes, which should be defined for all web pages. As part of publishing process, value of some of these attributes may be defined optionally; where as for other attributes it would be mandatory.

	Attribute	Type	Constraints	Description
1.	CreatorDeptName	String / Enum	Mandatory Max length 64	The Department primarily responsible for the intellectual content of the resource.
2.	CreatorOrgName	String	Optional Max length 64	The organization primarily responsible for the intellectual content of the resource.
3.	CreatorEmail	String / Enum	Mandatory Max length 64	The Contact person primarily responsible for the intellectual content of the resource.
4.	CoverageSpatial	String	Optional Geography Max length 128	Spatial topic and spatial applicability may be a named place or a location.
5.	CoverageTemporal	String	Optional Max length 128	Temporal topic may be a named period, date, or date range.
6.	CoverageJurisdiction	String	Mandatory Max length 128	A jurisdiction may be a named administrative entity or a geographic place to which the resource applies.
7.	Description	String	Mandatory Max length 1024	A textual description of the content of the resource, including abstracts in the case of document-like objects or content description in the case of visual resources.
8.	DateCreated	Date	Optional	The date the resource was made available in its original form
9.	DatePublished	Date	Mandatory	The date when the content item is published on web
10.	DateModified	Date	Optional	The date when the content item has been modified
11.	Format	String / Enum	Mandatory MIME/Type Take sub-set	The data representation of the resource, such as text/html, ASCII, Postscript file, executable application, or JPEG image. FORMAT will be assigned from enumerated lists such as registered Internet Media Types (MIME types). The MIME types are defined according to the RFC2046 standard. Currently, the only option available is the text/html option.
12.	Language	String / Enum	Mandatory Use subset of	Language of the intellectual content of the resource.

			ISO-639-2	
13.	PublisherOrgName	String	Optional Max length 128	The entity responsible for making the resource available in its present form, such as a publisher, a university department, or a corporate entity.
14.	PublisherDeptName	String / Enum	Mandatory Max length 128	Publisher department name
15.	PublisherEmail	String / Enum	Mandatory Max length 128	Publisher email
16.	PublisherAddress	String / Enum	Mandatory Max length 1024	Publisher Address
17.	PublisherPhone	String / Enum	Mandatory Max length 20	Publisher Phone number
18.	Relation	String []	Optional Max of 5 URLs Max URL length 128	The relationship to other resources. Recommended best practice is to identify the related resource by means of a string conforming to a formal identification system for example list of related URLs
19.	Source	String	Optional Max length 128	The work, either print or electronic, from which this resource is delivered, if applicable.
20.	Title	String	Mandatory Max length 128	The name given to the resource by the CREATOR or PUBLISHER
21.	TitleAlternate	String	Optional Max length 128	The Alternate name of content resource such as form16, act no, etc.
22.	SubjectKeywords	String	Mandatory Max length 1024 Separated by `;`	The topic of the resource, or keywords, phrases, or classification descriptors that describe the subject or content of the resource.
23.	SubjectClassification	String[] / Enum	Optional Max of 5 values Of Max length 128	Subject categories such as Audience , sectors, etc.
24.	Type	String / Enum	Mandatory	The category of the resource, such as ACTS, RULES, FORMS, DOCUMENTS, SERVICES, etc. It is expected that RESOURCE TYPE will be chosen from enumerated list of types.
25.	Identifier	String	Mandatory Max length 128	URL of the document/content

2.2. MDR Service Specifications

MDR Service would implement following interface.

```
public class MdrServiceStatusVO
{
    private boolean CMSStatus;
    private boolean DBStatus;
    private boolean MDRServiceStatus;
    private Calendar endTime;
    private Calendar replTime;
    private Calendar startTime;

    // Setters and getters .....
}

public final class ErrorCodes;
{
    // Following are error codes

    // Operation performed successfully
    public static final String    ERR_NO_ERROR = "ERR_NO_ERROR";

    // Arguments provided are invalid
    public static final String    ERR_INV_ARG = "ERR_INV_ARG";

    // One ore more metadata attributed are invalid
    public static final String    ERR_INV_METADATA = "ERR_INV_METADATA";

    // CMS is down or connection failed
    public static final String    ERR_CMS_DOWN = "ERR_CMS_DOWN";
}

public class MetadataSizeVO
{
    private int noOfNewWebPages;
    private int noOfModifiedWebPages;
    private int noOfExitedWebPages;
    private int totalNoOfAllWebPages;
    private long newMetadataSize;
    private long modifiedMetadataSize;
    private long exitedPagesSize;
    private long totalSizeOfAllMetadata;
    private String errCode, errDescr;

    private Calendar startTime;
    private Calendar endTime;
    private Calendar replTime;

    // Setters and getters .....
}

public class NotificationInfoVO
```



```
{
    private String eventDesc;
    private String errorNo;
    private String source;
    private String event_id;

    // Setters and getters .....
}

public class NotifyAckVO
{
    private boolean ackSts;
    private String errCode;
    private String errDesc;

    // Setters and getters .....
}

public class MetadataValidationErrVO
{
    private String errDescription=""; // Description of First error

    // all attributes that have error
    private String[] errAttributes =
        new String[MetadataProperties.AttributeOrder.values().length];
    private long metadataValidationErrVOSize=0;

    private int noOfErrorAttributes=0;// count of attributes that have error

    // Setters and getters .....
}
```

```
public class MetadataVO
{
    private String creatorDeptName;
    private String creatorOrgName;
    private String creatorEmail;
    private String coverageSpatial;
    private String coverageTemporal;
    private String coverageJurisdiction;
    private String description;
    private Calendar dateCreated;
    private Calendar datePublished;
    private Calendar dateModified;
    private String format;
    private String language;
    private String publisherOrgName;
    private String publisherDeptName;
    private String publisherEmail;
    private String publisherAddress;
    private String publisherPhone;
    private String[] relation= new String[MdcConstants.RELATIONS_ARRAY_SIZE];
    private String source;
    private String title;
    private String titleAlternate;
    private String subjectKeywords;
    private String[] subjectClassification=new
String[MdcConstants.SUBJ_CLASSIFICATION_ARRAY_SIZE];
    private String type; /** Identifier, which represents URL is not part of this class
*/

    // Setters and getters .....
}

public class WebPageMetadataVO
{
    /** Refers to the url of the web page */
    private String identifier="";

    /** State can be either published or exited */
    private String state;

    /** Represents MetadataValidation Value object */
    private MetadataValidationErrVO mdValidationErrVO=null;

    /** Represents reference to Metadata object */
    private MetadataVO metadata=null;

    // Setters and getters .....
}

public class WebPageMetadataSetVO
{
```

```
private WebPageMetadataVO[] webPegMetadataVOList;
private long sizeInBytes=0;
private int noErrWebPages=0;
private String errCode;
private String errDescr;
private Calendar startTime;
private Calendar endTime;
private Calendar replTime;
    // Setters and getters .....
}
```

```
public class ExitedWebpageSetVO
{
    private String[] exitedWebpagesList;
    private long sizeInBytes=0;
    private String errCode;
    private String errDescr;
    private Calendar startTime;
    private Calendar endTime;
    private Calendar replTime; // Setters and getters .....
}
```

```
public class WebSiteMetadataVO
{
    private long totalByteSize;
    private WebPageMetadataSetVO newlyCreatedPagesVO;
    private WebPageMetadataSetVO modifiedPagesVO;
    private ExitedWebpageSetVO exitedPagesVO;
    private String errCode;
    private String errDescr;
    private Calendar startTime;
    private Calendar endTime;
    private Calendar replTime;
        // Setters and getters .....
}
```

```

public interface IMdrService
{
    public MdrServiceStatusVO getStatus()throws java.rmi.RemoteException;

    public MetadataSizeVO
    getMetadataSize(Calendar startTime,Calendar    endTime)    throws
    java.rmi.RemoteException;

    public WebSiteMetadataVO
    getAllMetadata(Calendar startTime,Calendar    endTime)    throws
    java.rmi.RemoteException;

    public WebPageMetadataSetVO
    getNewMetadata(Calendar startTime,Calendar    endTime)    throws
    java.rmi.RemoteException;

    public WebPageMetadataSetVO
    getUpdatedMetadata(Calendar    startTime,Calendar    endTime)    throws
    java.rmi.RemoteException;

    public ExitedWebpageSetVO
    getExitedWebPages(Calendar startTime,Calendar    endTime)    throws
    java.rmi.RemoteException;

    public NotifyAckVO
    notifyWebSiteAdmin(NotificationInfoVO notificationInfo) throws
    java.rmi.RemoteException;

    public WebPageMetadataSetVO
    getWebpageMetadata(String[] webpageUrls) throws java.rmi.RemoteException;
}

```

Document/literal wrapped style WSDL, generated using this IMdrService interface, should be used for creating MDR service. ([Click here for download the WSDL file](#))

Following sub-sections explains each of the operations.

2.2.1. getStatus

	Item	Description
	Functionality	Return status of MDR service.
	Return value	Status of MDR service
	Input arguments	
	Void	

2.2.2. getWebPageMetadata

	Item	Description
	Functionality	Return metadata of the given webpage
	Return value	Webpage's metadata
	Input arguments	
	url[]	Arrays of URLs of web pages whose metadata is required.

2.2.3. getMetadataSize

	Item	Description
	Functionality	Return size of the metadata (new, modified and exited web pages), for the specified time period, which is required to be replicated.
	Return value	Size of metadata
	Input arguments	
	startTime	Start time of the time period
	endTime	End time of the time period

2.2.4. getAllMetadata

	Item	Description
	Functionality	Return all metadata (new, modified and exited pages) for the specified time period, which is required to be replicated.
	Return value	Website's metadata
	Input arguments	
	startTime	Start time of the time period
	endTime	End time of the time period

2.2.5. getNewMetadata

	Item	Description
	Functionality	Return new metadata created during the specified time period, which is required to be replicated.
	Return value	Web page metadata set
	Input arguments	
	startTime	Start time of the time period
	endTime	End time of the time period

2.2.6. getUpdatedMetadata

	Item	Description
	Functionality	Return modified metadata, modified during the specified time period, which is required to be replicated.
	Return value	Web page metadata set
	Input arguments	
	startTime	Start time of the time period
	endTime	End time of the time period

2.2.7. getExitedWebpages

	Item	Description
	Functionality	Return exited web pages during the specified time period, which is required to be replicated.
	Return value	Set of exited web pages
	Input arguments	
	startTime	Start time of the time period
	endTime	End time of the time period

2.2.8. notifyWebsiteAdmin

	Item	Description
	Functionality	Send specified notification to web site administrator.
	Return value	Notification acknowledgement
	Input arguments	
	NotificationInfoVO	Notification information

2.3. Message Level Security

MDC service should implement message level security based on username and password, which would be passed in SOAP message header as per "Web Services Security UsernameToken Profile 1.0" an OASIS standard (<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf>). MDC server uses WSS4J to implement message level security. It would include username and password parameters, in SOAP header. Sample SOAP request message is shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <soapenv:Header>
    <wsse:Security soapenv:mustUnderstand="1"
      xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-4536327"
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
        <wsse:Username>apmdrsrv</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordDigest">IWJHhAdSDClHz/SlfKQvaeKU5o=
        </wsse:Password>
        <wsse:Nonce>uXnrc7nD8vdyGbPLL/LLJA==</wsse:Nonce>
        <wsu:Created>2009-07-20T11:47:00.824Z</wsu:Created>
        </wsse:UsernameToken>
      </wsse:Security>
    </soapenv:Header>

    <soapenv:Body>
      <getStatus xmlns="http://mdc.npi.gov.in"/>
    </soapenv:Body>
  </soapenv:Envelope>
```

2.4. MDC-CMS-API Specifications

This interface encapsulates all communication with specific content repository. Its implementation depends on specific content repository or content management system. Typically implementation class makes use for software development kit of APIs provided by CMS product. Following is the definition of MSC-CMS-API interface.

```
public interface IMdcCmsApi
{
    public MdcCmsConnection connect (CmsConPropertiesVO cmsConProps,MetadataProperties metadataProp)
    throws MdcException,MetadataValidationException;

    public void disconnect (MdcCmsConnection conxn) throws MdcException;

    public void getAllMetadata (Calendar startTime, Calendar endTime, WebPageMetadataSet newMetadataSet,
    WebPageMetadataSet modifiedMetadataSet, WebPageMetadataSet exitedWebPageSet)
    throws MdcException;

    public void getNewMetadata (Calendar startTime, Calendar endTime, WebPageMetadataSet
    newMetadataSet) throws MdcException;

    public void getModifiedMetadata (Calendar startTime, Calendar endTime, WebPageMetadataSet
    modifiedMetadataSet) throws MdcException;

    public void getExitedWebPages (Calendar startTime, Calendar endTime, WebPageMetadataSet
    exitedWebPageSet) throws MdcException;

    public void getWebpageMetadata(String[] webPageUrls, WebPageMetadataSet urlWebPageSet )
    throws MdcException;
```



```
public void
startTransaction( )
throws MdcException;
```

```
public void
commitTransaction( )
throws MdcException;
```

```
public void
abortTransaction( )
throws MdcException;
}
```

Following sub-sections explain each of the operations. Refer source code file of definition of classes.

2.4.1. connect

	Item	Description
	Functionality	Establish connection with content repository or content management system.
	Return value	Connection object
	Input arguments	
	ConProp	Connection properties. Any specific properties required based on underlying content management system to implement MDC-CMS-API should be passed using this.
	mdProp	Metadata properties

2.4.2. disconnect

	Item	Description
	Functionality	Close given connection
	Return value	void
	Input arguments	
	conxn	Connection object

2.4.3. startTransaction

	Item	Description
	Functionality	Start transactions, all subsequent operations should be treated as part of this transaction
	Return value	void
	Input arguments	
	conxn	Connection object

2.4.4. commitTransaction

	Item	Description
	Functionality	Commit transaction, saving all modifications.
	Return value	void
	Input arguments	
	conxn	Connection object

2.4.5. abortTransaction

	Item	Description
	Functionality	Abort transaction without persisting modifications (if any)
	Return value	void
	Input arguments	
	conxn	Connection object

2.4.6. getWebpageMetadata

	Item	Description
	Functionality	Return metadata of web pages identified by given URL list.
	Return value	WebpageMetadataSet: Metadata of given web pages. If web page is exited then metadata would not be returned.
	Input arguments	
	urlList	List of web pages (URLs) whose metadata is required
	wpMetada	Metadata of given web pages

2.4.7. getAllMetadata

	Item	Description
	Functionality	Return all metadata (new, modified and exited web pages) for the specified time period.
	Return value	void
	Input arguments	
	startTime	Start time of the time period
	endTime	End time of the time period
	Output arguments	
	newMetadata	Metadata of new web pages
	modMetadata	Metadata of modified web pages
	exitedPages	List of exited pages

2.4.8. getNewMetadata

	Item	Description
	Functionality	Return metadata of web pages created during the

		specified time period.
	Return value	void
	Input arguments	
	startTime	Start time of the time period
	endTime	End time of the time period
	Output arguments	
	newMetadata	Metadata of new web pages

2.4.9. getUpdatedMetadata

	Item	Description
	Functionality	Return metadata of web pages, which are modified during the specified time period.
	Return value	void
	Input arguments	
	startTime	Start time of the time period
	endTime	End time of the time period
	Output arguments	
	modMetadata	Metadata of modified web pages

2.4.10. getExitedWebpages

	Item	Description
	Functionality	Return list of web pages, which are existed during the specified time period.
	Return value	void
	Input arguments	
	startTime	Start time of the time period
	endTime	End time of the time period
	Output arguments	
	exitedPages	List of exited pages

3. MDR Service

MDR service is a web service. It would be deployed on each of the websites. Its behavior would be controlled using following property files.

Metadata.xml : Metadata attribute properties

MdrService.xml : MDR service properties

3.1. MdrService.xml

Following table explains the attributes of MdrService.xml property file.

	Attribute Name	Type	Value Range	Description
1	AdminEmailId	String	Max length 256	Website administrator's email IDs (separated by `;` or `;`)
2	EscalationEmailId	String	Max length 256	Email IDs (separated by `;` or `;`) to whom escalation email would be send.
3	WebsiteBaseURL	String	Max length 64	Base URL of website. This is used by MDC-CMS-API
4	EmailServer	Group		Email server connection properties
	EmailServer->HostName	String	Max length 64	Host name of email server
	EmailServer->MdrSrvEmailId	String	Max length 128	Email ID on whose behalf MDR service would send email
	EmailServer->UserID	String	Max length 64	User Id for authenticating to email server
	EmailServer->Password	String	Max length 64	Password for authenticating to email server
5	DataServer	Group		Database server connection properties
	DataServer->ConnectionArg	String	Max length 128	Arguments or parameters for connecting to database server
	DataServer->UserID	String	Max length 64	User id for authenticating to database server
	DataServer->Password	String	Max length 64	Password for authenticating to database server
	DataServer->DriverClassName	String	Max length 128	Database driver class
6	ContentRepository	Group		
	ContentRepository->RepositoryID	String	Max length 128	ID of content repository
	ContentRepository->WebSiteId	String	Max length 128	Project ID of website within content repository
	ContentRepository->UserID	String	Max length 64	

	ContentRepository->Password	String	Max length 64	
7	EventHandler	Group		Actions to be taken for each event
	EventHandler->EventId	String	One from the defined list	Event ID of the action. Following are the events DB_DOWN CMS_DOWN MDR_SERVICE_DOWN RECV_INVALID_MD MD_VALIDATION_ERR SYSTEM_ERROR MD_REPL_FAILED MD_REPL_SUCCESS MD_REPL_FAILURE_ESCLTN DOWN_TIME_ESCLTN
	EventHandler->ActionSet	Group		
	EventHandler->ActionSet->Action	String	One from the defined list	Action ID. Following is the list of actions. LOG2FILE LOG2CONSOLE LOG2SYSLOG SEND_NOTIFICATION

Note: Based on underlying content management system and implementation of publishing workflow some more properties may be added to MdrService.xml property file.

[XML schema and sample property files of MdrService](#) (MdrService.xsd, MdrService.xml)

3.2. Metadata.xml

Following table explains the attributes of Metadata.xml property file.

	Attribute Name	Type	Value Range / Description
1	CreatorOrgName	Group	
	CreatorOrgName->Mandatory	Boolean	TRUE / FALSE
	CreatorOrgName->MaxLength	Integer	1 to 64
2	CreatorDeptProp	Group	
	CreatorDeptProp->Mandatory	Boolean	TRUE / FALSE
	CreatorDeptProp->CreatorDept	Group	Enum: List of creator departments
	CreatorDeptProp->CreatorDept->Name	String	Name of creator department
	CreatorDeptProp->CreatorDept->Email	String	Email Id of creator department
3	CoverageSpatial	Group	
	CoverageSpatial->Mandatory	Boolean	TRUE / FALSE
	CoverageSpatial->MaxLength	Integer	1 to 128

	Attribute Name	Type	Value Range / Description
4	CoverageTemporal	Group	
	CoverageTemporal->Mandatory	Boolean	TRUE / FALSE
	CoverageTemporal->MaxLength	String	1 to 128
5	CoverageJurisdiction	Group	
	CoverageJurisdiction->Mandatory	Boolean	TRUE / FALSE
	CoverageJurisdiction->MaxLength	String	1 to 128
6	Description	Group	
	Description->Mandatory	Boolean	TRUE / FALSE
	Description->MaxLength	Integer	1 to 1024
7	DateCreated	Group	
	DateCreated->Mandatory	Boolean	TRUE / FALSE
8	DatePublished	Group	
	DatePublished->Mandatory	Boolean	TRUE / FALSE
9	DateModified	Group	
	DateModified->Mandatory	Boolean	TRUE / FALSE
10	Format	Group	
	Format->Mandatory	Boolean	TRUE / FALSE
	Format->FormatID	String	Enum: List of valid file formats
11	Language	Group	TRUE / FALSE
	Language->Mandatory	Boolean	TRUE / FALSE
	Language->LanguageID	String	Enum: List of languages
12	PublisherOrgname	Group	
	PublisherOrgName->Mandatory	Boolean	TRUE / FALSE
	PublisherOrgName->MaxLength	Integer	1 to 128
13	PublisherDeptProp	Group	
	PublisherDeptProp->Mandatory	Boolean	TRUE / FALSE
	PublisherDeptProp->Publisher	Group	Enum: List of publisher department
	PublisherDeptProp->Publisher->Name	String	Name of publisher department
	PublisherDeptProp->Publisher->Email	String	Email Id of publisher department
	PublisherDeptProp->Publisher->Address	String	Address of publisher department
	PublisherDeptProp->Publisher->Phone	String	Phone number publisher department
14	Relation	Group	
	Relation->Mandatory	Boolean	TRUE / FALSE
	Relation->MaxURLS	Integer	1 to 5

	Attribute Name	Type	Value Range / Description
	Relation->MaxUrlLength	Integer	1 to 128
15	Source		
	Source->Mandatory	Boolean	TRUE / FALSE
	Source->MaxLength	Integer	1 to 128
16	Title	Group	
	Title->Mandatory	Boolean	TRUE / FALSE
	Title->MaxLength	Integer	1 to 128
17	TitleAlternate	Group	
	TitleAlternate->Mandatory	Boolean	TRUE / FALSE
	TitleAlternate->MaxLength	Integer	1 to 128
18	SubjectKeyWords		
	SubjectKeyWords->Mandatory	Boolean	TRUE / FALSE
	SubjectKeyWords->MaxLength	Integer	1 to 128
19	SubjectClassification	Group	
	SubjectClassification->Mandatory	Boolean	TRUE / FALSE
	SubjectClassification->SubjectID	String	Enum: List of categories
20	Type	Group	
	Type->Mandatory	Boolean	TRUE / FALSE
	Type->TypeID	String	Enum: List of resource types

[XML schema and sample property files of Metadata.](#)

4. MDC Server

MDC server consists of two jobs namely status monitor and metadata consolidator. Below sections explains each one of this.

4.1. MDR Service Status Monitor

Monitoring of MDR services would be achieved using this batch job, which would be executed periodically using operating system provided scheduling tool. It would make use a property file "MdcServer.xml" to control its behavior.

4.1.1. MdcServer.xml

Following table explains the attributes of MdcServer.xml property file.

	Attribute Name	Type	Value Range	Description
1	MaxRoundTripTime	Int	1 to 64K	Maximum time rime in minutes for round trip of an MDR service operation. Based on this metadata would be retrieved either per event or

	Attribute Name	Type	Value Range	Description
				all at a time.
2	ChunkTimeDuration	Integer	1 to 64k	Time duration in minutes, for retrieving metadata in small durations.
3	EscalationDurationMD	Integer	1 to 64K	Time in minutes, after which if metadata replication error persists, escalation should take place
4	EscalationDurationStatus	Integer	1 to 64K	Time in minutes, after which if MDR service remains down, escalation should take place
5	AdminEmailId	String	Max length 256	MDC Server's administrator's email IDs (separated by ';' or ',')
6	EscalationEmailId	String	Max length 256	MDC server's email IDs (separated by ';' or ',') to whom escalation email would be send.
7	EmailServer	Group		Email server connection properties
	EmailServer->HostName	String	Max length 64	Host name of email server
	EmailServer->MdcSrvEmailId	String	Max length 64	Email ID on whose behalf MDC server would send email
	EmailServer->UserID	String	Max length 64	User Id for authenticating to email server
	EmailServer->Password	String	Max length 64	Password for authenticating to email server
8	DataServer	Group		Database server connection properties
	DataServer->ConnectionArg	String	Max length 128	Arguments or parameters for connecting to database server
	DataServer->UserID	String	Max length 64	User id for authenticating to database server
	DataServer->Password	String	Max length 64	Password for authenticating to database server
	DataServer->DriverClassName	String	Max length 128	Database driver class
9	WebsiteEmailId	Group		Website administrator's email ID list
	WebsiteEmailId -> WebsiteId	String	Max length 32	Web site ID
	WebsiteEmailId->AdminEmailId	String	Max length 256	Email IDs (separated by ';' or ',') of website administrators
	WebsiteEmailId->EscalationEmailId	String	Max length 256	Email IDs (separated by ';' or ',') for escalation
10	EventHandler	Group		Actions to be taken for each event
	EventHandler->	String	One from the	Event ID of the action.

	Attribute Name	Type	Value Range	Description
	EventId		list	Following are the events DB_DOWN CMS_DOWN MDR_SERVICE_DOWN RECV_INVALID_MD MD_VALIDATION_ERR SYSTEM_ERROR MD_REPL_FAILED MD_REPL_SUCCESS MD_REPL_FAILURE_ESCLTN DOWN_TIME_ESCLTN MDR_SRV_RECOVERED MD_REPL_RECOVERED
	EventHandler-> ActionSet	Group		
	EvebtHandler-> ActionSet->Action	String	One from the list	Action ID. Following is the list of actions. LOG2FILE LOG2CONSOLE LOG2SYSLOG SEND_NOTIFICATION
11	MdrService	Group		
	MdrService-> WebSiteID	String	Max length 128	Name/ID of website
	MdrService-> EndpointURL	String	Max length 128	End point URL of MDR service
	MdrService-> UserID	String	Max length 64	
	MdrService-> Password	String	Max length 64	

[XML schema and sample property files of MDCserver.](#)

4.2. Metadata Consolidator

This batch job would be responsible for retrieving metadata from MDR services. It would be executed periodically using operating system provided scheduling tool. It would make use of a property files to control its behavior.

Metadata.xml: Metadata attribute properties. This is same as explained in previous section

MdcServer.xml: MDC server properties. This is same as explained in previous section

5. MDC-CMS-API Implementation

This class implements MDC-CMS-API based on underlying WCMS product. It would encapsulate all communication with content repository. It makes use of some of the properties passed from MDR service to control its behavior.

6. Checklist for MDR Service

	Checklist Item	Remarks
1	Is specified WSDL file is used to generate stubs for MDR service?	
2	Is MDR service implementation is based on stubs generated from the specified WSDL file?	
3	Is MDC-CMS-API interfaces implemented to encapsulate all functionality related to communication with content repository?	
4	Is standard metadata.xsd file used to validate metadata.xml property file?	
5	Is metadata.xml property defined as per defined standard?	
6	Is standard MdrService.xsd file used to validate MdrService.xml property file?	
7	Is MdrService.xml property file defined as per defined standard?	
8	Is message level security based on "Web Services Security UsernameToken Profile 1.0" implemented?	