



Framework Corporatiu J2EE

Connector Documentum

Versió 1.2

Barcelona, 09 / febrer / 2007



Històric de modificacions

Data	Autor	Comentaris	Versió
09/02/2007	Atos Origin	Actualització de la documentació Versió 1.2 d'OpenFrame	1.2
6/02/2006	Atos Origin, sae openTrends	Versió inicial del document	1.0
03/10/2006	Atos Origin	Afegir exemple d'ús	1.0.1

Llegenda de Marcadors



Índex

1.	INTRODUCCIÓ	5
1.1.	PROPÓSIT	5
1.2.	CONTEXT I ESCENARIS D'ÚS	5
1.3.	VERSIONS I DEPENDÈNCIES	5
1.3.1.	<i>Versions</i>	5
1.3.2.	<i>Dependències Bàsiques</i>	6
1.4.	A QUI VA DIRIGIT	6
1.5.	DOCUMENTS I FONTS DE REFERÈNCIA	6
1.6.	GLOSSARI	6
2.	DESCRIPCIÓ DETALLADA	7
2.1.	ARQUITECTURA I COMPONENTS	7
2.1.1.	<i>Interfícies i Components Genèrics</i>	7
	Session i SessionImpl	9
	SessionManager i SessionManagerImpl	10
	DocumentumService i DocumentumServiceImpl	11
	DocumentumConfig	12
	DocumentumConnectorService	14
	DocumentumConnectorImpl	19
2.1.2.	<i>Components basats en Commons i Spring</i>	26
2.2.	INSTAL·LACIÓ I CONFIGURACIÓ	27
2.2.1.	<i>Instal·lació</i>	27
2.2.2.	<i>Configuració</i>	35
2.3.	UTILITZACIÓ DEL SERVEI	37
2.3.1.	<i>Creació fitxer propietats</i>	37
2.3.2.	<i>Creació bean d'Spring</i>	38
2.3.3.	<i>Injecció del bean en el BO</i>	38
2.3.4.	<i>Inici sessió documentum i ús</i>	39
2.4.	INTEGRACIÓ AMB ALTRES SERVEIS	40
2.4.1.	<i>Integració amb el Servei de Internacionalització</i>	40



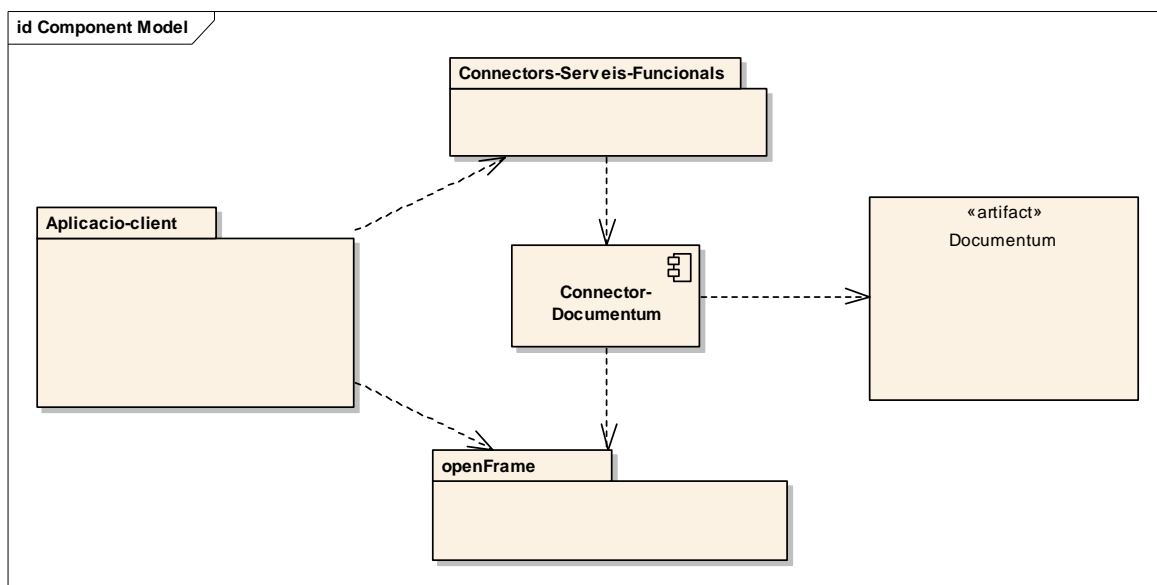
1. Introducció

1.1. Propòsit

El propòsit del connector és proporcionar una interfície Java per accedir a Documentum. Permetent emmagatzemar i recuperar documents a més d'altres operacions relacionades com crear carpetes i associar propietats.

1.2. Context i Escenaris d'Ús

El connector documentum es troba dins els connectors de serveis funcional.



El connector documentum es troba dins els Connectors de Serveis Funcionals.

1.3. Versions i Dependències

En el present apartat es mostren quines són les versions i dependències necessàries per fer ús del Servei.

1.3.1. Versions

Respecte la versió anterior s'han introduït diagrames de classes així com la signatura de les diferents operacions de les diferents classes juntament amb una breu descripció.



1.3.2. Dependències Bàsiques

Nom	Tipus	Versió	Descripció
commons-logging	jar	1.0.4	http://jakarta.apache.org/commons
dfc	jar	1.0	
dfcbase	jar	1.0	
junit	jar	3.8.1	
log4j	jar	1.2.12	
openFrame-services-exceptions	jar	1.0.1	
openFrame-services-logging	jar	1.0.1	
spring	jar	1.2.5	http://www.springframework.org
xml-apis	jar	1.3.02	

1.4. A qui va dirigit

Aquest document va dirigit als següents perfils:

- Programador. Per conèixer l'ús del connector.
- Arquitecte. Per conèixer quins són els components i la configuració del connector.

1.5. Documents i Fonts de Referència

- | | | |
|-----|--|---|
| [1] | Anàlisi Funcional - Connector
Serveis Funcionals - 16-Nov.doc | Document d'anàlisi dels connectors de serveis funcionals. |
| [2] | Documentum Foundation Classes
(DFC) | http://www.dmdeveloper.com/articles/dfc/ |

1.6. Glossari



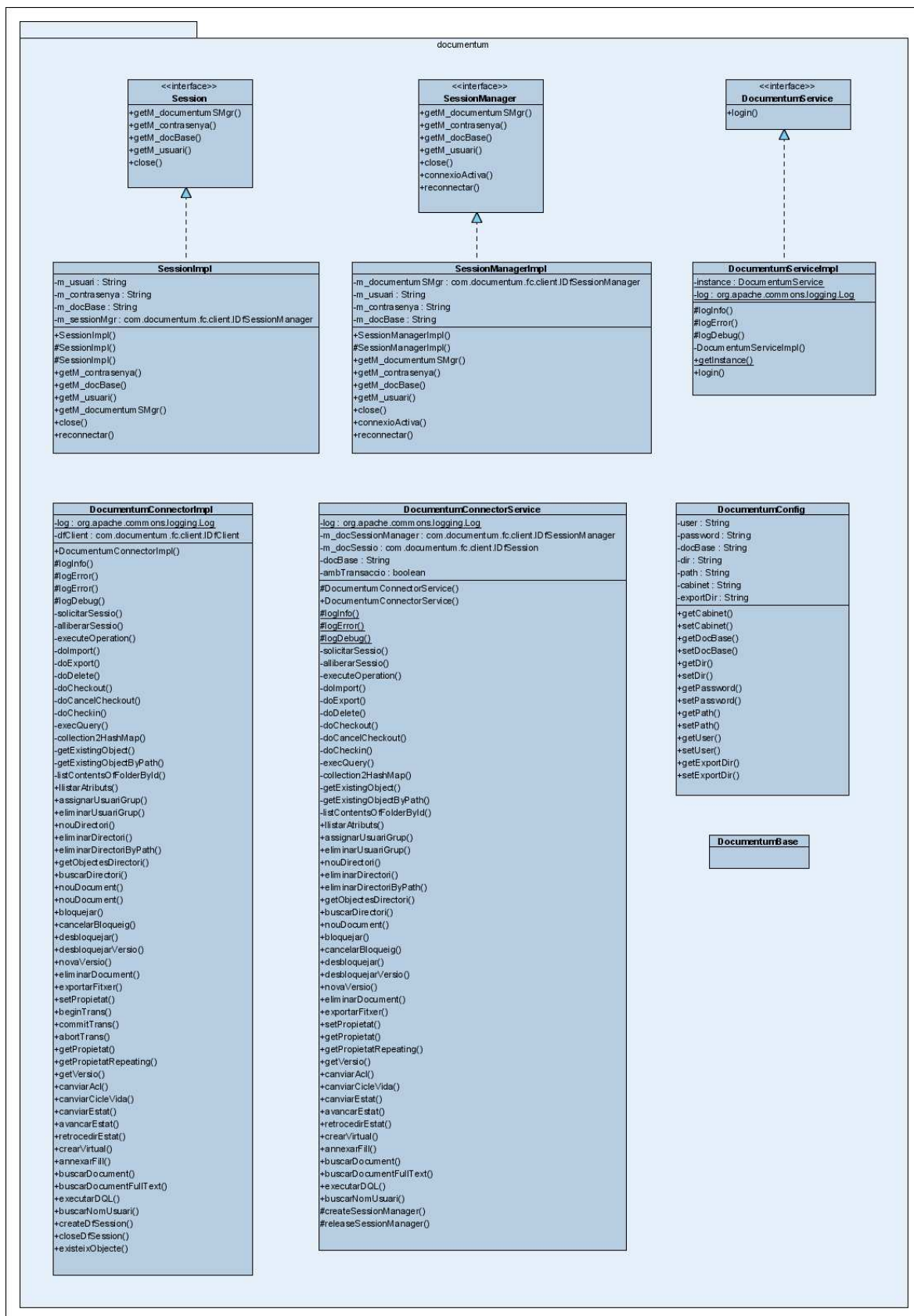
2. Descripció Detallada

2.1. Arquitectura i Components

2.1.1. Interfícies i Components Genèrics

Les principals interfícies i classes que les implementen són:

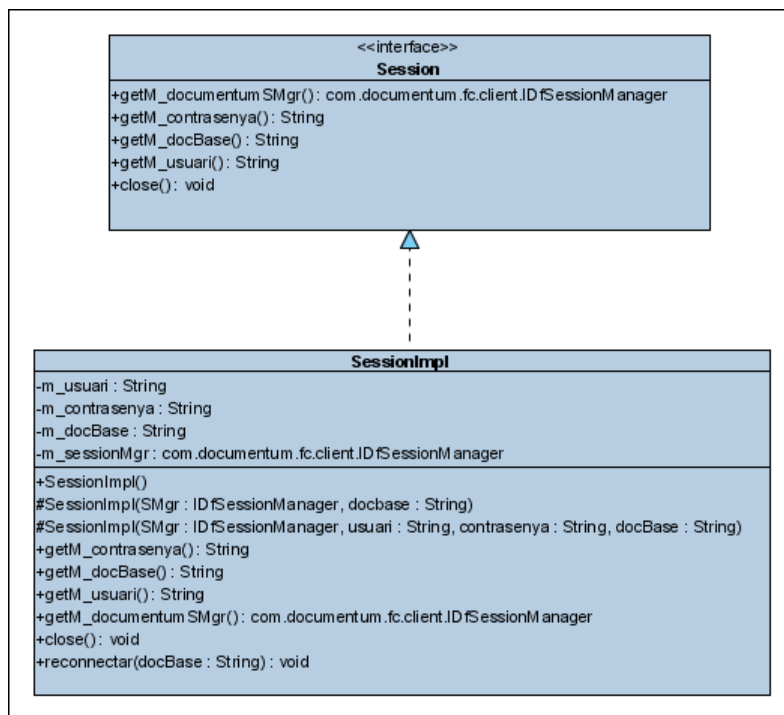
Component		Package	Descripció
Interfície	Implementació		
DocumentumService	DocumentumServiceImpl	net.soluziona.documen tum	Per obtenir accés el servei. (login per iniciar sessió)
Session	SessionImpl	net.soluziona.documen tum	Sessió amb la cual interactuar amb Documentum
SessionManager	SessionManagerImpl	net.soluziona.documen tum	S'utilitza internament (per obtenir una sessió)
DocumentumConne ctorService	DocumentumConnectorI mpl	net.soluziona.documen tum	S'utilitza internament (per alliberar una sessió)





Session i SessionImpl

La interfície **Session**, implementada per **SessionImpl**, ofereix un gestor de sessions d'un usuari a Documentum.



Signatura de les operacions:

```
public IDfSessionManager getM_documentumSMgr();
```

```
public String getM_contrasenya();
```

```
public String getM_docBase();
```

```
public String getM_usuari();
```

```
public void close() throws DocumentumException;
```

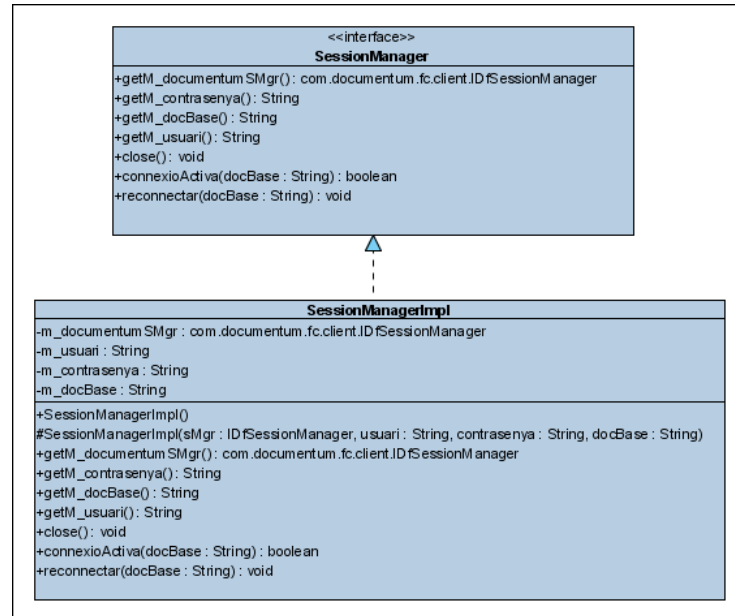
Tanca la sessió de documentum.

```
public void reconnectar(String docBase) {
```



SessionManager i SessionManagerImpl

La interfície **SessionManager**, implementada per **SessionManagerImpl**, ofereix un gestor de sessions d'un usuari a Documentum.



Signatura de les operacions:

public IDfSessionManager **getM_documentumSMgr()**;
Obté el SessionManager.

public String **getM_contrasenya()**;
Obté el password per connectar a Documentum.

public String **getM_docBase()**;
Obté el nom del repositori documental o DocBase.

public String **getM_usuari()**;
Obté l'usuari per accedir a Documentum.

public void **close()** throws DocumentumException;
Tanca la sessió de Documentum.

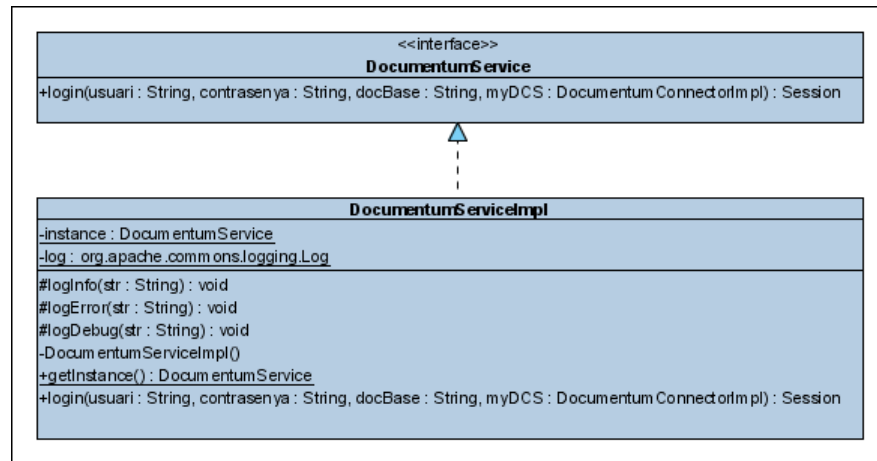
public boolean **connexioActiva**(String docBase) throws DocumentumException;
Valida que la connexió amb Documentum estigui activa.
docBase: nom del repositori documental o DocBase a comprovar que és activa.

public void **reconnectar**(String docBase) {
Reactiva una connexió prèvia.
docBase: nom del repositori documental o DocBase amb la que es reactiva una connexió prèvia.



DocumentumService i DocumentumServiceImpl

La interfície **DocumentumService**, implementada per **DocumentumServiceImpl**, ofereix a través de la creació d'una classe **SessionManager** l'autenticació (**login**) i abandonar de sessió (**logout**) a Documentum.



Signatura de les operacions:

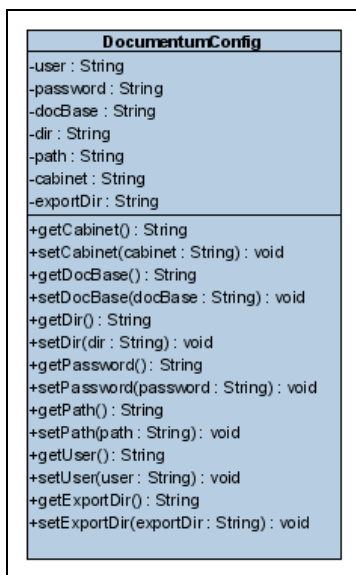
public static DocumentumService **getInstance** ();
Obté una única instància de la classe

public Session **login** (String usuari, String contrasenya, String docBase, DocumentumConnectorImpl myDCS)
throws DocumentumException;
Obre una nova sessió o recull una previa ja existent per al usuari
usuari: nom d'usuari per accedir a la base de documents o repositori documental.
contrasenya: password de l'usuari per accedir a la base de documents o repositori documental.
DocBase: nom del repositori documental o DocBase.
myDCS: connector de Documentum.
Retorna un objecte **Session**.



DocumentumConfig

La classe **DocumentumConfig** permet emmagatzemar tota la informació necessària per establir la connexió amb Documentum.



Signatura de les operacions:

public String **getCabinet()**;

Obté el cabinet indicat per la connexió amb Documentum.
Retorna un String amb el nom del cabinet.

public void **setCabinet**(String cabinet);

Posa el cabinet per la connexió amb Documentum.
cabinet: cabinet per la connexió amb Documentum.

public String **getDocBase()**;

Obté la base de documents (nom del repositori documental o DocBase) indicat per la connexió amb Documentum.
Retorna un String amb el nom del repositori documental o DocBase.

public void **setDocBase**(String docBase) ;

Posa la base de documents (nom del repositori documental o DocBase) per la connexió amb Documentum.
docBase: nom del repositori documental o DocBase.

public String **getDir()**;

Obté el directori indicat per la connexió amb Documentum.
Retorna un String amb el directori del repositori documental o DocBase.

public void **setDir**(String dir) ;

Posa el directori per la connexió amb Documentum.
dir: directori del repositori documental o DocBase.



public String **getPassword**();

Obté el password indicat per la connexió amb Documentum.

Retorna un String amb el password del repositori documental o DocBase.

public void **setPassword**(String password) ;

Posa el password per la connexió amb Documentum.

password: password per la connexió amb el repositori documental o DocBase.

public String **getPath**();

Obté el path indicat per la connexió amb Documentum.

Retorna un String amb el path del repositori documental o DocBase.

public void **setPath**(String path) ;

Posa el path per la connexió amb Documentum.

path: path el repositori documental o DocBase.

public String **getUser**();

Obté l'usuari indicat per la connexió amb Documentum.

Retorna un String amb el user amb del repositori documental o DocBase.

public void **setUser**(String user) ;

Posa l'usuari per la connexió amb Documentum.

user: nom de l'usuari del repositori documental o DocBase.

public String **getExportDir**();

Obté el directori d'exportació indicat per la connexió amb Documentum.

Retorna un String amb el directori d'exportació.

public void **setExportDir**(String exportDir) ;

Posa el directori d'exportació per la connexió amb Documentum.

exportDir: directori en el que s'exportaran els documents del repositori documental o DocBase.



DocumentumConnectorService

La classe **DocumentumConnectorService** implementa un connector amb Documentum 5.x via DFCs.

DocumentumConnectorService
<pre>-log : org.apache.commons.logging.Log -m_docSessionManager : com.documentum.fc.client.IDfSessionManager -m_docSessio : com.documentum.fc.client.IDfSession -docBase : String -ambTransaccio : boolean #DocumentumConnectorService() +DocumentumConnectorService(sMgr : SessionManager) #logInfo(str : String) : void #logError(str : String) : void #logDebug(str : String) : void -solicitarSessio() : void -alliberarSessio() : void -executeOperation(operation : com.documentum.operations.IDfOperation) : void -dolImport(srcFileOrDir : String, destFolderPath : String) : String -doExport(sysObj : IDfSysObject, destDir : String, exportName : String, exportFormat : String) : void -doDelete(sysObj : IDfSysObject) : void -doCheckout(sysObj : IDfSysObject) : void -doCancelCheckout(sysObj : IDfSysObject) : void -doCheckin(sysObj : IDfSysObject, pathFixerSrc : String, checkinVersion : int) : void -execQuery(queryString : String) : IDfCollection -collection2HashMap(col : IDfCollection) : HashMap -getExistingObject(strObjId : String) : IDfSysObject -getExistingObjectByPath(strPath : String) : IDfSysObject -listContentsOfFolderByld(strFolderId : String) : IDfCollection +listarAtributs(tipus : String) : void +assignarUsuariGrup(grup : String, usuari : String) : void +eliminarUsuariGrup(grup : String, usuari : String) : void +nouDirectorio(path : String, nom : String, tipus : String) : void +eliminarDirectorio(idDirectorio : String) : void +eliminarDirectorioByPath(path : String) : void +getObjectesDirectorio(idDirectorio : String) : HashMap +buscarDirectorio(resultats : String, tipus : String, camps : String, text : String, path : String, exacte : boolean, ordre : String) : HashMap +nouDocument(ad : String, dominiAcl : String, srcPath : String, dstPath : String, tipusObj : String) : String +bloquejar(idDocument : String) : void +cancelarBloqueig(idDocument : String) : void +desbloquejar(pathFixer : String, idDocument : String) : void +desbloquejarVersio(pathFixer : String, idDocument : String) : void +novaVersio(pathFixer : String, idDocument : String) : void +eliminarDocument(idDocument : String) : void +exportarFixer(dstPath : String, idDocument : String) : void +setPropietat(idObjecte : String, propietat : String, valor : String) : void +getPropietat(idObjecte : String, propietat : String) : String +getPropietatRepeating(idObjecte : String, propietat : String) : String +getVersio(idObjecte : String) : String +canviarAcl(idObjecte : String, nomACL : String, dominiACL : String) : void +canviarCicleVida(idObjecte : String, estat : String, nomCicleVida : String) : void +canviarEstat(idObjecte : String, estat : String) : void +avancarEstat(idObjecte : String) : void +retrocedirEstat(idObjecte : String) : void +crearVirtual(idDocument : String) : void +annexarFill(idDocumentFill : String, idDocument : String) : void +buscarDocument(resultats : String, tipus : String, camps : String, text : String, path : String, exacte : boolean, ordre : String) : HashMap +buscarDocumentFullText(resultats : String, tipus : String, text : String, path : String, ordre : String) : HashMap +executarDDL(sDDL : String) : HashMap +buscarNomUsuari(usuariLogin : String) : String #createSessionManager(usuari : String, contrasenya : String, docBase : String) : IDfSessionManager #releaseSessionManager(sMgr : IDfSessionManager) : void</pre>



public **DocumentumConnectorService**(SessionManager sMgr);

Crea un nou objecte DocumentumConnectorService fent ús del SessionManager, responsable de la connexió amb el DocBase de Documentum.

sMgr: SessionManager per establir la connexió amb el repositori documental de Documentum.

public void **listarAtributs**(String tipus) **throws** DocumentumException;

Ens permet conèixer els atributs d'un tipus a Documentum.

tipus: tipus de documents dels que volem conèixer els atributs.

public void **assignarUsuariGrup**(String grup, String usuari) ;

Assignar un usuari a un grup a partir del seu grup a la DocBase i del nom de l'usuari.

grup: grup de Documentum al que volem afegir l'usuari.

usuari: usuari de Documentum que volem afegir al grup.

public void **eliminarUsuariGrup**(String grup, String usuari) ;

Desassociar un usuari d'un grup de la DocBase.

grup: grup de Documentum del que volem esborrar l'usuari.

usuari: usuari de Documentum que volem esborrar del grup.

public void **nouDirectori**(String path, String nom, String tipus);

Crea un nou directori o cabinet a la DocBase, per això, es necessita:

path: si es crea un directori cal indicar la ubicació del directori; si es crea un cabinet aquest paràmetre es deixarà en blanc.

nom: nom del directori o cabinet.

tipus: tipus intern: "dm_folder" o "dm_cabinet".

public void **eliminarDirectori**(String idDirectori);

Eliminar un directori a partir del seu identificador.

idDirectori: directori a esborrar de Documentum.

public void **eliminarDirectoriByPath**(String path) **throws** DocumentumException;

Elimina un directori a partir de la seva ubicació del directori a la DocBase.

path: path del directori a esborrar de Documentum.

public HashMap **getObjectesDirectori**(String idDirectori);

Obté tots els objectes d'un directori a partir del identificador del directori.

idDirectori: identificador de directori.

public HashMap **buscarDirectori**

(String resultats, String tipus, String camps, String text, String path, boolean exacte, String ordre)

throws DocumentumException;

Realitza una recerca en un directori de la DocBase a partir dels següents paràmetres:

resultats: camps que volem incloure en la recerca.

tipus: taula sobre la qual volem fer la recerca.

camps: camps pels quals volem filtrar la recerca.

text: text a comparar amb els camps de la recerca.

path: indica el punt de partida de la recerca.

exacte: indica si volen filtrar el camp text de manera exacta.

ordre: camps indicant l'ordre de retorn de la recerca.

Retorna un HashMap<Integer, Vector>



public String **nouDocument**
(String acl, String dominiAcl, String srcPath, String dstPath, String tipusObj)
throws DocumentumException;
Crea un nou document a la DocBase.
acl: ACL del document.
dominiAcl: domini ACL del document.
srcPath: path on trobem el document a Local.
dstPath: path on s'ubicarà el document a la DocBase.
nomObj: nom del document a Local.
tipusObj: tipus intern "dm_document" o algun que heredi d'aquest
Retorna el identificador intern del document.

public void **bloquejar**(String idDocument) **throws** DocumentumException;
Fa un check-out del document.
idDocument: Identificador del document.

public void **cancelarBloqueig**(String idDocument) **throws** DocumentumException;
Cancel·la el check-out.
idDocument: identificador del document.

public void **desbloquejar**(String pathFitxer, String idDocument) **throws** DocumentumException;
Fa un check-in sense incrementar la versió del document.
pathFitxer (Opcional): Indica l'adreça a local on podem trobar el fitxer.
idDocument: Identificador del document.

public void **desbloquejarVersio**(String pathFitxer, String idDocument) **throws** DocumentumException;
Fa un check-in incrementant la versió del document.
pathFitxer (Opcional): Indica l'adreça a local on podem trobar el fitxer.
idDocument: Identificador del document

public void **novaVersio**(String pathFitxer, String idDocument) **throws** DocumentumException;
Fa un check-out i un check-in augmentant la versió del document.
pathFitxer (Opcional): Indica l'adreça a local on podem trobar el fitxer.
idDocument: Identificador del document.

public void **eliminarDocument**(String idDocument) **throws** DocumentumException;
Elimina un document.
idDocument: Identificador del document.

public void **setPropietat**(String idObjecte, String propietat, String valor) **throws** DocumentumException;
Modifica les propietats d'un objecte.
idObjecte: Identificador de l'objecte.
propietat Nom de la propietat.
valor: Nou valor de la propietat.

public String **getPropietat**(String idObjecte, String propietat) **throws** DocumentumException;
idObjecte: Identificador de l'objecte.
propietat: Nom de la propietat a mostrar la informació.
Retorna el valor de la propietat.



public String **getPropietatRepeating**(String idObjecte, String propietat) **throws** DocumentumException;

Obté les propietats d'un objecte de tipus repeating.

idObjecte: Identificador de l'objecte.

propietat: Nom de la propietat a mostrar la informació.

Retorna el valor de la propietat separats per " | " (pipes).

public String **getVersio**(String idObjecte) **throws** DocumentumException;

Mostra la versió actual d'un document.

idObjecte: Identificador del document.

Retorna la versió actual.

public void **canviarAcl**(String idObjecte, String nomACL, String dominiACL) **throws** DocumentumException;

Canvia el ACL d'un document.

idObjecte: Identificador de l'objecte.

nomACL: Nom del ACL.

dominiACL: Domini del ACL.

public void **canviarCicleVida**(String idObjecte, String estat, String nomCicleVida)

throws DocumentumException;

Canviem el cicle de vida d'un objecte.

idObjecte: Identificador de l'objecte.

estat: Estat inicial del nou cicle de vida.

nomCicleVida: Nom del cicle de vida.

public void **canviarEstat**(String idObjecte, String estat) **throws** DocumentumException;

Canviem l'estat del cicle de vida d'un objecte.

idObjecte: Identificador de l'objecte.

estat: Nou estat de l'objecte.

public void **retrocedirEstat**(String idObjecte) **throws** DocumentumException;

Retrocedeix un estat del cicle de vida d'un document

idObjecte: Identificador del document.

public void **avancarEstat**(String idObjecte) **throws** DocumentumException;

Avança un estat del cicle de vida d'un document

idObjecte: Identificador del document.

public void **crearVirtual**(String idDocument) **throws** DocumentumException;

Modificat l'estat d'un document per a fer-lo virtual.

idDocument: Identificador del document.

public void **annexarFill**(String idDocumentFill, String idDocument) **throws** DocumentumException;

Annexar un fill a un document virtual.

idDocumentFill: identificador del fill.

idDocument: identificador del pare.



public HashMap **buscarDocument**

(String resultats, String tipus, String camps, String text, String path, boolean exacte, String ordre)

throws DocumentumException;

A partir d'una serie de paràmetres d'entrada realitza una recerca a la DocBase per buscar documents.

resultats: Camps que volem inclou-re en al recerca.

tipus: Taula sobre la qual volem realitzar la recerca.

camps: Camps pels quals volem filtrar la recerca.

text: Text a comparar amb els camps de la recerca.

path: Indica el punt de partida dela recerca.

exacte: Indicati de si volen filtrar el camp text de manera exacte.

ordre: Camps indicant l'ordre de retorn la recerca.

Retorna un HashMap<Integer, Vector>

public HashMap **buscarDocumentFullText**

(String resultats, String tipus, String text, String path, String ordre) **throws** DocumentumException;

A partir d'una serie de paràmetres d'entrada realitza una recerca a la DocBase per buscar documents.

resultats: Camps que volem inclou-re en al recerca.

tipus: Taula sobre la qual volem realitzar la recerca.

text: Text a comparar amb els camps de la recerca.

path: Indica el punt de partida dela recerca.

ordre: Camps indicant l'ordre de retorn la recerca.

Retorna un HashMap<Integer, Vector>.

public HashMap **executarDQL**(String sDQL) **throws** DocumentumException;

Executa una DQL i retorna els resultats.

sDQL: DQL a executar.

Retorna un HashMap<Integer, Vector>

public String **buscarNomUsuari**(String usuariLogin) **throws** DocumentumException;

Retorna el nom d'usuari (user_name) a partir del login (user_login_name).

usuariLogin: Login Usuari

Retorna un String



DocumentumConnectorImpl

La classe **DocumentumConnectorImpl** implementa un connector amb Documentum 5.x via DFCs.

DocumentumConnectorImpl
<pre>-log : org.apache.commons.logging.Log -dfClient : com.documentum.fc.client.IDfClient +DocumentumConnectorImpl() #logInfo(str : String) : void #logError(str : String) : void #logError(str : String, ex : com.documentum.fc.common.DfException) : void #logDebug(str : String) : void -solicitarSessio(session : Session) : IDfSession -alliberarSessio(sMgr : IDfSessionManager, dfSession : IDfSession) : void -executeOperation(operation : IDfOperation) : void -dolImport(srcFileOrDir : String, destFolderPath : String, pObjType : String, pName : String, m_docSessio : IDfSession) : String -doExport(sysObj : IDfSysObject, destDir : String, exportName : String, exportFormat : String) : String -doDelete(sysObj : IDfSysObject) : void -doCheckout(sysObj : IDfSysObject, dfSession : IDfSession) : void -doCancelCheckout(sysObj : IDfSysObject) : void -doCheckin(sysObj : IDfSysObject, pathFixerSrc : String, checkinVersion : int) : void -execQuery(queryString : String, m_docSessio : IDfSession) : IDfCollection -collection2HashMap(col : IDfCollection) : HashMap -getExistingObject(strObjId : String, m_docSessio : IDfSession) : IDfSysObject -getExistingObjectByPath(strPath : String, m_docSessio : IDfSession) : IDfSysObject -listContentsOfFolderById(strFolderId : String, m_docSessio : IDfSession) : IDfCollection +llistarAtributs(tipus : String, sessio : Session) : void +assignarUsuariGrup(grup : String, usuari : String, sessio : Session) : void +eliminarUsuariGrup(grup : String, usuari : String, sessio : Session) : void +nouDirector(path : String, nom : String, tipus : String, sessio : Session) : void +eliminarDirector(idDirector : String, sessio : Session) : void +eliminarDirectorByPath(path : String, sessio : Session) : void +getObjectesDirector(idDirector : String, sessio : Session) : HashMap +buscarDirector(resultats : String, tipus : String, camps : String, text : String, path : String, exacte : boolean, ordre : String, sessio : Session) : HashMap +nouDocument(ad : String, dominiAcl : String, srcPath : String, dstPath : String, tipusObj : String, sessio : Session) : String +nouDocument(ad : String, dominiAcl : String, srcPath : String, dstPath : String, pName : String, tipusObj : String, sessio : Session) : String +bloquejar(idDocument : String, sessio : Session) : void +cancelarBloqueig(idDocument : String, sessio : Session) : void +desbloquejar(pathFixer : String, idDocument : String, sessio : Session) : void +desbloquejarVersio(pathFixer : String, idDocument : String, sessio : Session) : void +novaVersio(pathFixer : String, idDocument : String, sessio : Session) : void +eliminarDocument(idDocument : String, sessio : Session) : void +exportarFixer(dstPath : String, idDocument : String, sessio : Session) : String +setPropietat(idObjecte : String, propietat : String, valor : String, sessio : Session) : void +beginTrans(sessio : Session) : void +commitTrans(sessio : Session) : void +abortTrans(sessio : Session) : void +getPropietat(idObjecte : String, propietat : String, sessio : Session) : String +getPropietatRepeating(idObjecte : String, propietat : String, sessio : Session) : String +getVersio(idObjecte : String, sessio : Session) : String +canviarAcl(idObjecte : String, nomACL : String, dominiACL : String, sessio : Session) : void +canviarCicleVida(idObjecte : String, estat : String, nomCicleVida : String, sessio : Session) : void +canviarEstat(idObjecte : String, estat : String, sessio : Session) : void +avancarEstat(idObjecte : String, sessio : Session) : void +retrocedirEstat(idObjecte : String, sessio : Session) : void +crearVirtual(idDocument : String, sessio : Session) : void +annexarFill(idDocumentFill : String, idDocument : String, sessio : Session) : void +buscarDocument(resultats : String, tipus : String, camps : String, text : String, path : String, exacte : boolean, ordre : String, sessio : Session) : HashMap +buscarDocumentFullText(resultats : String, tipus : String, text : String, path : String, ordre : String, sessio : Session) : HashMap +executarDQL(sDQL : String, sessio : Session) : HashMap +buscarNomUsuari(usuariLogin : String, sessio : Session) : String +createDfSession(usuari : String, contrasenya : String, docBase : String) : IDfSessionManager +closeDfSession(sMgr : IDfSessionManager) : void +existeixObjecte(strObjId : String, sessio : Session) : boolean</pre>



public void **llistarAtributs**(String tipus, Sessio sessio) **throws** DocumentumException;

Mètode utilitzat per conèixer els atributs d'un tipus a Documentum.

tipus: Nom del tipus.

sessio: Sessió establerta amb Documentum.

public void **assignarUsuariGrup**(String grup, String usuari, Sessio sessio);

Assignar un usuari a un grup a partir del seu grup a la DocBase i del nom de l'usuari.

grup: grup de Documentum al que volem afegir l'usuari.

usuari: usuari de Documentum que volem afegir al grup.

sessio: Sessió establerta amb Documentum.

public void **eliminarUsuariGrup**(String grup, String usuari, Sessio sessio);

Desassociar un usuari d'un grup de la DocBase.

grup: grup de Documentum del que volem esborrar l'usuari.

usuari: usuari de Documentum que volem esborrar del grup.

sessio: Sessió establerta amb Documentum.

public void **nouDirectori**(String path, String nom, String tipus, Sessio sessio);

Crea un nou directori o cabinet a la DocBase, per això, es necessita:

path: si es crea un directori cal indicar la ubicació del directori; si es crea un cabinet aquest paràmetre es deixarà en blanc.

nom: nom del directori o cabinet.

tipus: tipus intern: "dm_folder" o "dm_cabinet".

sessio: Sessió establerta amb Documentum.

public void **eliminarDirectori**(String idDirectori, Sessio sessio);

Eliminar un directori a partir del seu identificador.

idDirectori: directori a esborrar de Documentum.

sessio: Sessió establerta amb Documentum.

public void **eliminarDirectoriByPath**(String path, Sessio sessio) **throws** DocumentumException;

Elimina un directori a partir de la seva ubicació del directori a la DocBase.

path: path del directori a esborrar de Documentum.

sessio: Sessió establerta amb Documentum.

public HashMap **getObjectesDirectori**(String idDirectori, Sessio sessio);

Obté tots els objectes d'un directori a partir del identificador del directori.

idDirectori: identificador de directori.

sessio: Sessió establerta amb Documentum.

public HashMap **buscarDirectori**

(String resultats, String tipus, String camps, String text, String path, boolean exacte, String ordre, Sessio sessio) **throws** DocumentumException;

Realitza una recerca en un directori de la DocBase a partir dels següents paràmetres:

resultats: camps que volem incloure en la recerca.

tipus: taula sobre la qual volem fer la recerca.

camps: camps pels quals volem filtrar la recerca.

text: text a comparar amb els camps de la recerca.

path: indica el punt de partida de la recerca.

exacte: indica si volen filtrar el camp text de manera exacta.

ordre: camps indicant l'ordre de retorn de la recerca.

sessio: Sessió establerta amb Documentum

Retorna un HashMap<Integer, Vector>



public String **nouDocument**

(String acl, String dominiAcl, String srcPath, String dstPath, String tipusObj, Sessio sessio)

throws DocumentumException;

Crea un nou document a la DocBase.

acl: ACL del document.

dominiAcl: domini ACL del document.

srcPath: path on trobem el document a Local.

dstPath: path on s'ubicarà el document a la DocBase.

nomObj: nom del document a Local.

tipusObj: tipus intern "dm_document" o algun que heredi d'aquest

sessio: Sessió que s'ha establert amb Documentum.

Retorna el identificador intern del document.

public String **nouDocument**

(String acl, String dominiAcl, String srcPath, String dstPath, String pName, String tipusObj,

Sessio sessio) **throws** DocumentumException;

Crea un nou document a la DocBase.

acl: ACL del document.

dominiAcl: domini ACL del document.

srcPath: path on trobem el document a Local.

dstPath: path on s'ubicarà el document a la DocBase.

nomObj: nom del document a Local.

pName: nom del document.

tipusObj: tipus intern "dm_document" o algun que heredi d'aquest

sessio: Sessió que s'ha establert amb Documentum.

Retorna el identificador intern del document.

public void **bloquejar**(String idDocument, Sessio sessio) **throws** DocumentumException;

Fa un check-out del document.

idDocument: Identificador del document.

sessio: Sessió que s'ha establert amb Documentum.

public void **cancelarBloqueig**(String idDocument, Sessio sessio) **throws** DocumentumException;

Cancel·la el check-out.

idDocument: identificador del document.

sessio: Sessió que s'ha establert amb Documentum.

public void **desbloquejar**(String pathFitxer, String idDocument, Sessio sessio) **throws** DocumentumException;

Fa un check-in sense incrementar la versió del document.

pathFitxer (Opcional): Indica l'adreça a local on podem trobar el fitxer.

idDocument: Identificador del document.

sessio: Sessió que s'ha establert amb Documentum.

public void **desbloquejarVersio**(String pathFitxer, String idDocument, Sessio sessio)

throws DocumentumException;

Fa un check-in incrementant la versió del document.

pathFitxer (Opcional): Indica l'adreça a local on podem trobar el fitxer.

idDocument: Identificador del document

sessio: Sessió que s'ha establert amb Documentum.



public void **novaVersio**(String pathFitxer, String idDocument, Session sessio) **throws** DocumentumException;

Fa un check-out i un check-in augmentant la versió del document.

pathFitxer (Opcional): Indica l'adreça a local on podem trobar el fitxer.

idDocument: Identificador del document.

sessio: Sessió que s'ha establert amb Documentum.

public void **eliminarDocument**(String idDocument) **throws** DocumentumException;

Elimina un document.

idDocument: Identificador del document.

sessio: Sessió que s'ha establert amb Documentum.

public String **exportarFitxer**(String dstPath, String idDocument, Session sessio)

throws DocumentumException;

Exporta un fitxer de la DocBase a disc local.

dstPath: path de la màquina local on ubicar el document.

idDocument: Identificador del document.

sessio: Sessió que s'ha establert amb Documentum.

Retorna un String amb el path del document exportat.

public void **setPropietat**(String idObjecte, String propietat, String valor, Session sessio)

throws DocumentumException;

Modifica les propietats d'un objete.

idObjecte: Identificador de l'objecte.

propietat Nom de la propietat.

valor: Nou valor de la propietat.

sessio: Sessió que s'ha establert amb Documentum.

public void **beginTrans**(Session sessio) ;

Inicia una transacció.

sessio: Sessió que s'ha establert amb Documentum.

public void **commitTrans** (Session sessio) ;

Fa commit de la transacció.

sessio: Sessió que s'ha establert amb Documentum.

public void **abortTrans** (Session sessio) ;

Aborta la transacció.

sessio: Sessió que s'ha establert amb Documentum.

public String **getPropietat**(String idObjecte, String propietat, Session sessio) **throws** DocumentumException;

idObjecte: Identificador de l'objecte.

propietat: Nom de la propietat a mostrar la informació.

sessio: Sessió que s'ha establert amb Documentum.

Retorna el valor de la propietat.



public String **getPropietatRepeating**(String idObjecte, String propietat, Session sessio) **throws** DocumentumException;

Obté les propietats d'un objecte de tipus repeating.

idObjecte: Identificador de l'objecte.

propietat: Nom de la propietat a mostrar la informació.

sessio: Sessió que s'ha establert amb Documentum.

Retorna el valor de la propietat separats per " | " (pipes).

public String **getVersio**(String idObjecte, Session sessio) **throws** DocumentumException;

Mostra la versió actual d'un document.

idObjecte: Identificador del document.

sessio: Sessió que s'ha establert amb Documentum.

Retorna la versió actual.

public void **canviarAcl**(String idObjecte, String nomACL, String dominiACL, Session sessio) **throws** DocumentumException;

Canvia el ACL d'un document.

idObjecte: Identificador de l'objecte.

nomACL: Nom del ACL.

dominiACL: Domini del ACL.

sessio: Sessió que s'ha establert amb Documentum.

public void **canviarCicleVida**(String idObjecte, String estat, String nomCicleVida, Session sessio) **throws** DocumentumException;

Canviem el cicle de vida d'un objecte.

idObjecte: Identificador de l'objecte.

estat: Estat inicial del nou cicle de vida.

nomCicleVida: Nom del cicle de vida.

sessio: Sessió que s'ha establert amb Documentum.

public void **canviarEstat**(String idObjecte, String estat, Session sessio) **throws** DocumentumException;

Canviem l'estat del cicle de vida d'un objecte.

idObjecte: Identificador de l'objecte.

estat: Nou estat de l'objecte.

sessio: Sessió que s'ha establert amb Documentum.

public void **retrocedirEstat**(String idObjecte, Session sessio) **throws** DocumentumException;

Retrocedeix un estat del cicle de vida d'un document

idObjecte: Identificador del document.

sessio: Sessió que s'ha establert amb Documentum.

public void **avancarEstat**(String idObjecte, Session sessio) **throws** DocumentumException;

Avança un estat del cicle de vida d'un document

idObjecte: Identificador del document.

sessio: Sessió que s'ha establert amb Documentum.

public void **crearVirtual**(String idDocument, Session sessio) **throws** DocumentumException;

Modifica l'estat d'un document per a fer-lo virtual.

idDocument: Identificador del document.

sessio: Sessió que s'ha establert amb Documentum.



public void **annexarFill**(String idDocumentFill, String idDocument, Session sessio)
throws DocumentumException;

Annexar un fill a un document virtual.

idDocumentFill: identificador del fill.

idDocument: identificador del pare.

sessio: Sessió que s'ha establert amb Documentum.

public HashMap **buscarDocument**

(String resultats, String tipus, String camps, String text, String path, boolean exacte, String ordre, Session sessio) **throws** DocumentumException;

A partir d'una serie de paràmetres d'entrada realitza una recerca a la DocBase per buscar documents.

resultats: Camps que volem inclou-re en al recerca.

tipus: Taula sobre la qual volem realitzar la recerca.

camps: Camps pels quals volem filtrar la recerca.

text: Text a comparar amb els camps de la recerca.

path: Indica el punt de partida dela recerca.

exacte: Indicati de si volen filtrar el camp text de manera exacte.

ordre: Camps indicant l'ordre de retorn la recerca.

sessio: Sessió que s'ha establert amb Documentum.

Retorna un HashMap<Integer, Vector>

public HashMap **buscarDocumentFullText**

(String resultats, String tipus, String text, String path, String ordre, Session sessio)

throws DocumentumException;

A partir d'una serie de paràmetres d'entrada realitza una recerca a la DocBase per buscar documents.

resultats: Camps que volem inclou-re en al recerca.

tipus: Taula sobre la qual volem realitzar la recerca.

text: Text a comparar amb els camps de la recerca.

path: Indica el punt de partida dela recerca.

ordre: Camps indicant l'ordre de retorn la recerca.

sessio: Sessió que s'ha establert amb Documentum.

Retorna un HashMap<Integer, Vector>.

public HashMap **executarDQL**(String sDQL, Session sessio) **throws** DocumentumException;

Executa una DQL i retorna els resultats.

sDQL: DQL a executar.

sessio: Sessió que s'ha establert amb Documentum.

Retorna un HashMap<Integer, Vector>

public String **buscarNomUsuari**(String usuariLogin, Session sessio) **throws** DocumentumException;

Retorna el nom d'usuari (user_name) a partir del login (user_login_name).

usuariLogin: Login Usuari

sessio: Sessió que s'ha establert amb Documentum.

Retorna un String

public IdfSessionManager **createDfSession**(String usuari, String contrasenya, String docBase)

throws DocumentumException;

Sol licita un SessionManager a una DocBase.

usuari: nom de l'usuari.

contrasenya: contrasenya de l'usuari.

docBase: DocBase a la que s'ha de connectar.

Retorna un element IdfSessionManager amb el pool de connexions.



public void **closeDfSession**(IdfSessionManager sMgr);

Allibera un SessionManager a una DocBase.

sMgr: SessionManager a alliberar.

public boolean **existeixObjecte**(String strObjId, Session sessio) **throws** DocumentumException;

Comprova si l'objecte existeix amb aquest identificador.

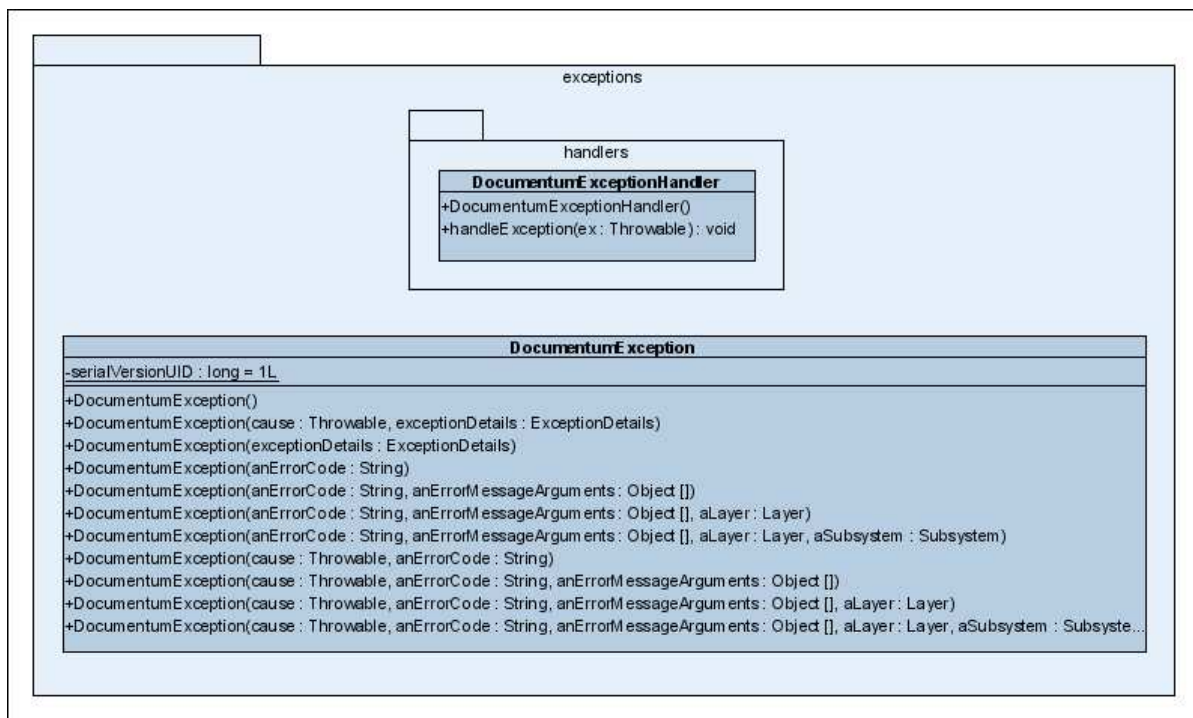
strObjId: identificador de l'objecte.

sessio: Sessió que s'ha establert amb Documentum.

Retorna un boolean indicant si existeix algun objecte a Documentum amb aquell identificador.

2.1.2. Components basats en Commons i Spring

Component	Package	Descripció
DocumentumException	net.soluziona.documentum.exceptions	S'utilitza per a la gestió d'errors
DocumentumExceptionHandler	net.soluziona.documentum.exceptions.handlers	S'utilitza per a la gestió d'errors

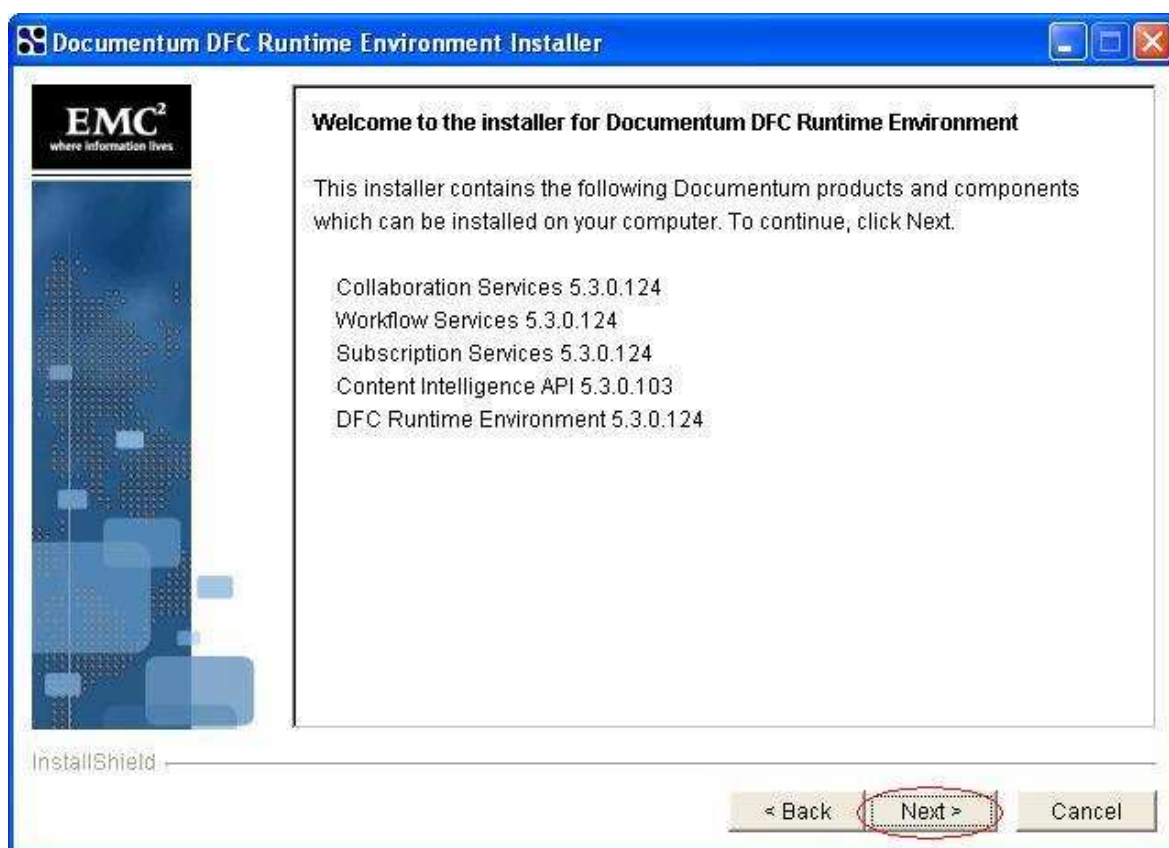


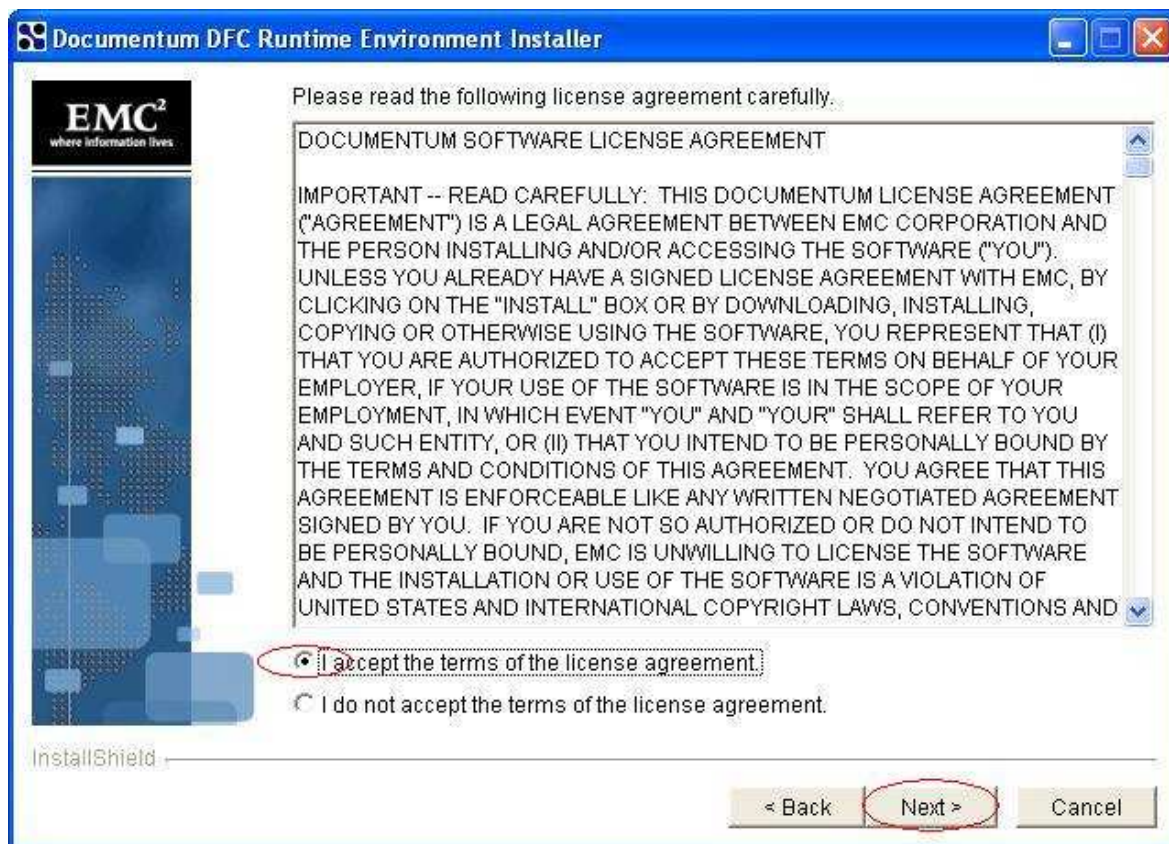
2.2. Instal·lació i Configuració

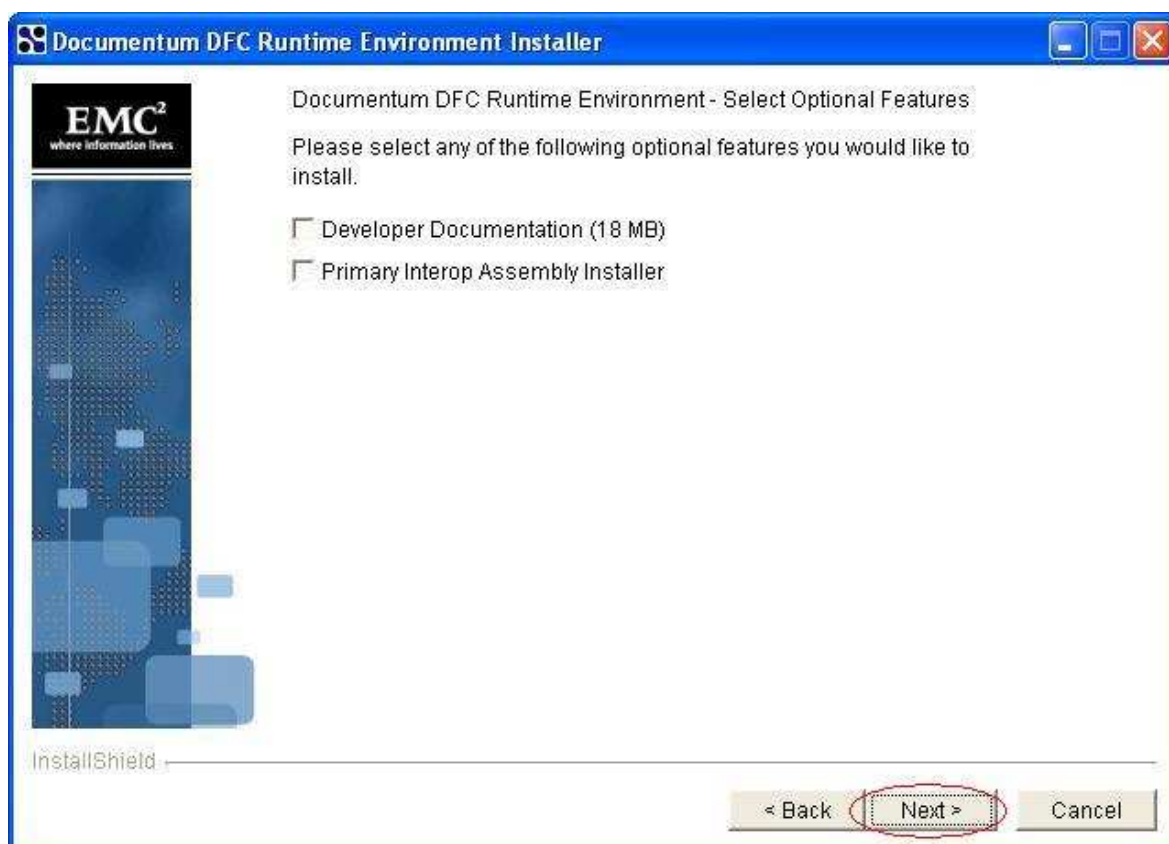
2.2.1. Instal·lació

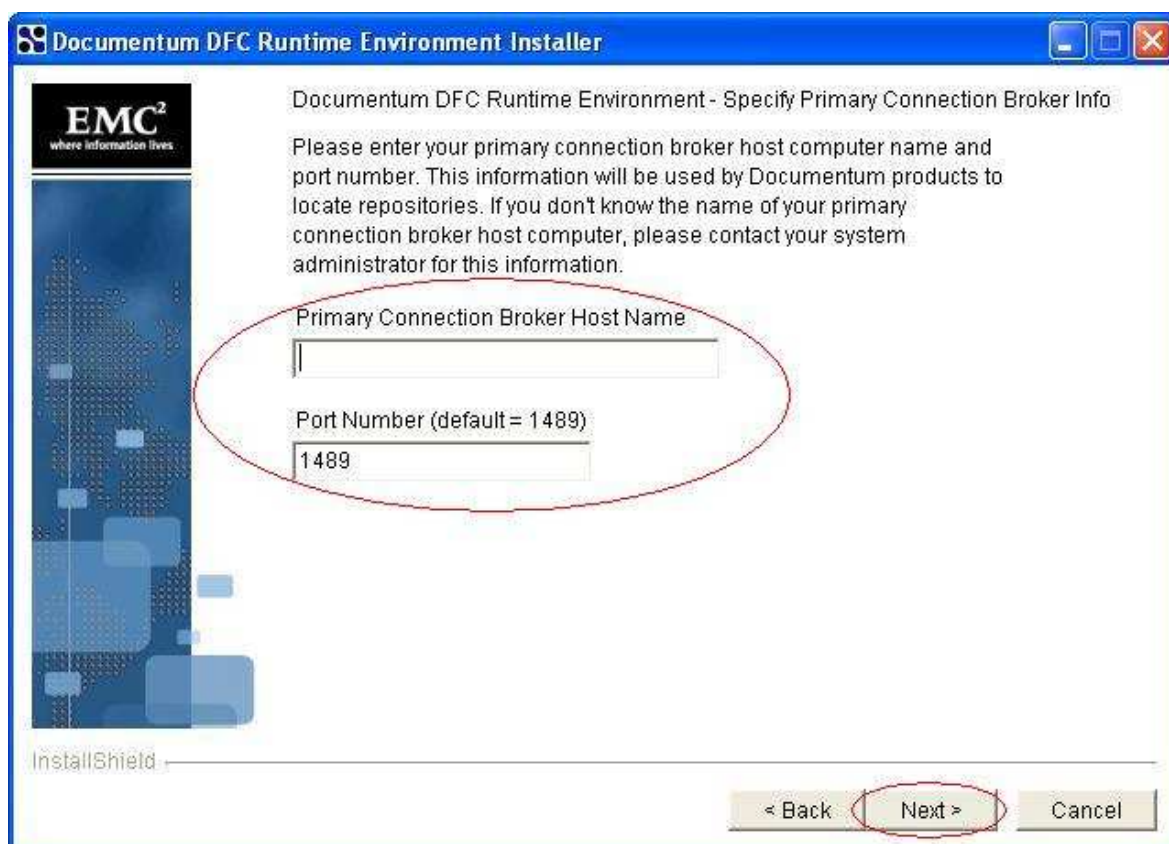
La instal·lació del connector requereix de la utilització de la llibreria 'openFrame-connectors-documentum' i les dependències indicades a l'apartat 'Introducció-Versions i Dependències'.

Ademés s'han de instal·lar les llibreries *DFC_5.3_SP1_windows.exe* que formen part del producte Documentum. Els passos a seguir en l'instal·lació són els següents:









Aquí indicarem el host i el port de documentum. Aquesta part crearà el fitxer C:\WINDOWS\dmcl.ini, que és on guardarà paràmetres com el host i el port. En aquest fitxer podrem canviar aquesta configuració un cop finalitzada la instal·lació. Tindrà el següent format:

[DOCBROKER_PRIMARY]

host =es-hg2r02j

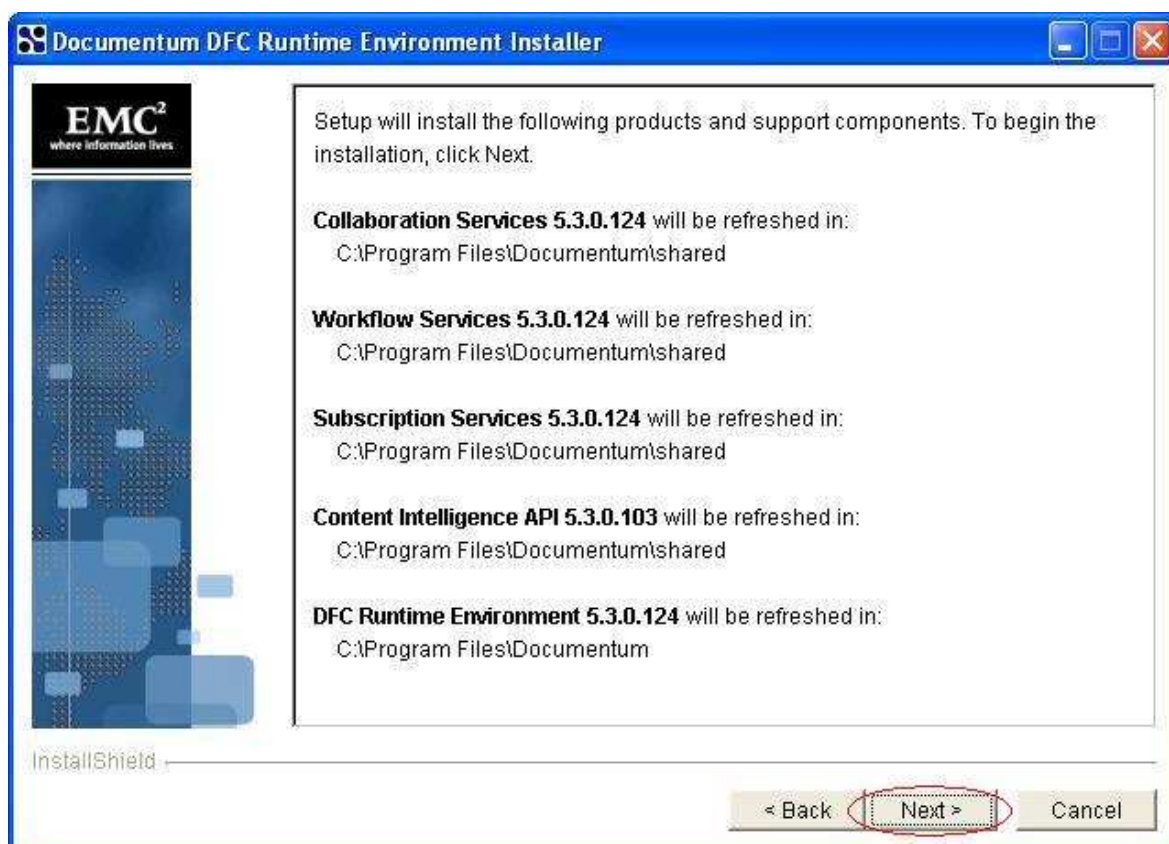
port =1489

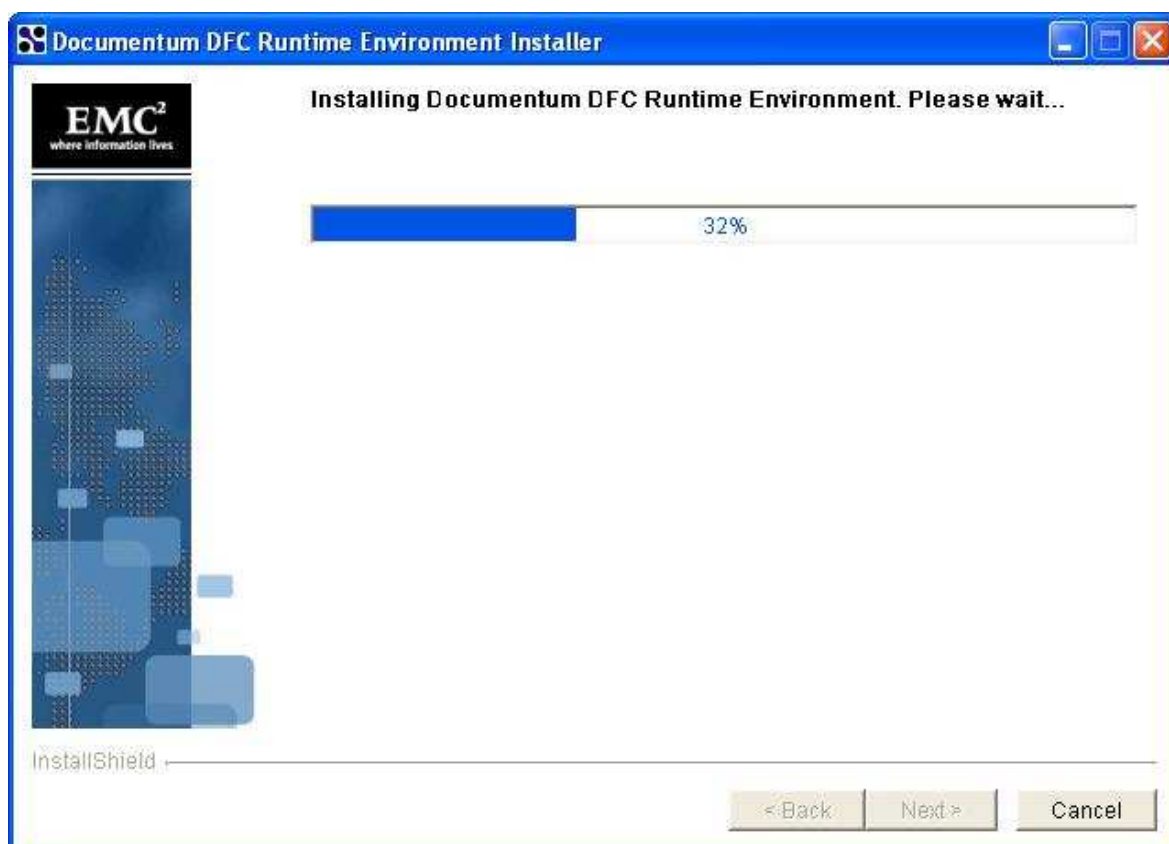
[DMAPI_CONFIGURATION]

cache_queries = T

token_storage_enabled=F

token_storage_path=D:\Documentum\apptoken







Documentum DFC Runtime Environment Installer

EMC²
where information lives

Documentum DFC Runtime Environment - Designate Global Registry

Designate which repository is the global registry for this installation.

☒ Designate the global registry repository to use

Repository Name (must be version 5.3 or later):

Login Name:

Password:

☒ Test Connection

InstallShield

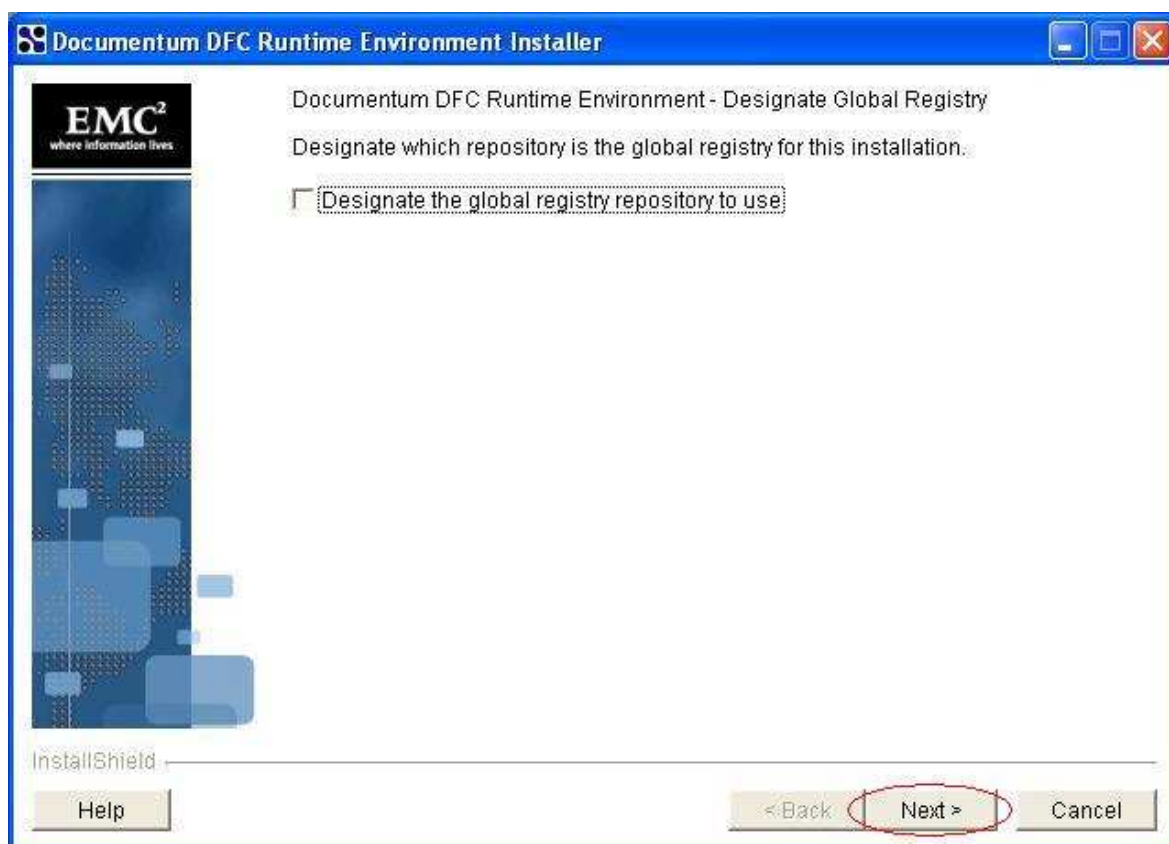
Help < Back Next > Cancel

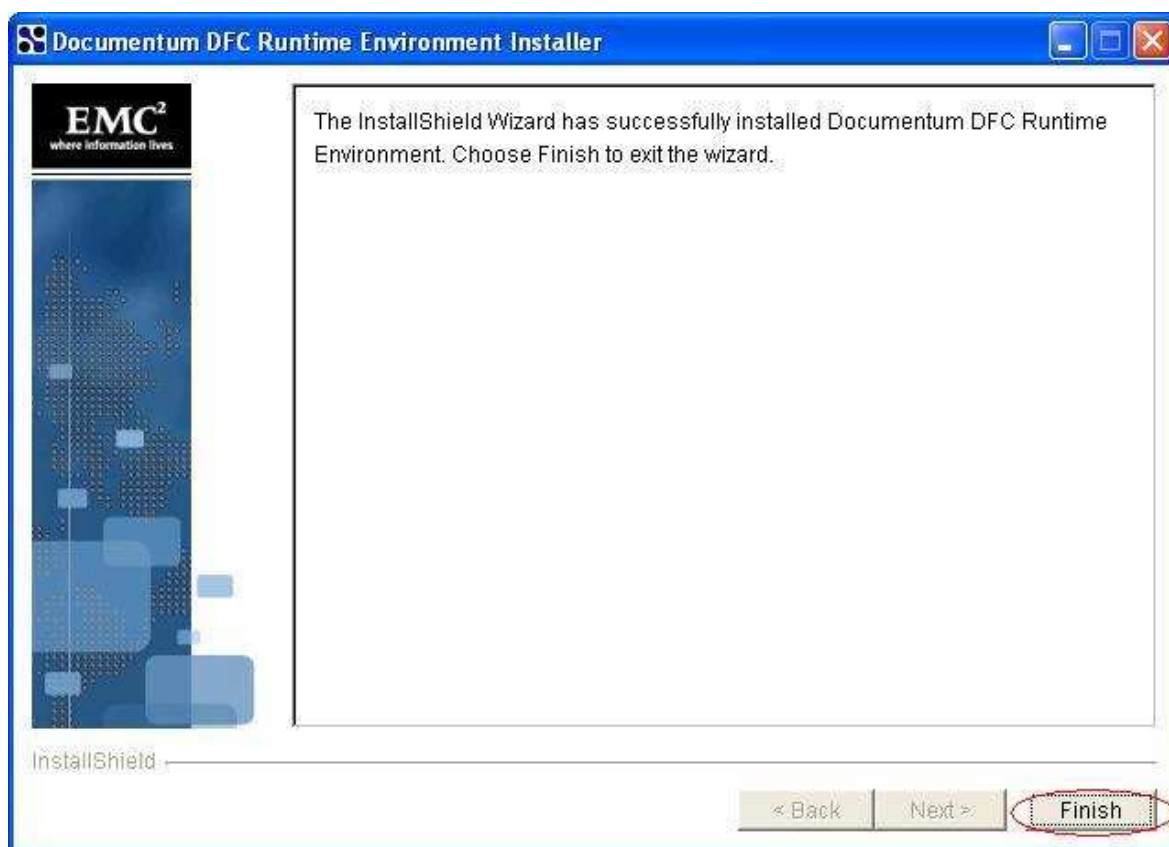
WARNING

Valid global registry connection information is required for Collaborative Services and using network locations, a distributed content feature.

For instructions on how to designate and enable a global registry at a later time, please review the Help panel.

OK





2.2.2. Configuració

Serà necessari el fitxer de configuració de la pròpia aplicació que utilitza el connector documentum on inclourem l'utilització del connector.

Exemple:

```
<bean name="documentumConnectorBean"  
class="net.soluziona.documentum.DocumentumConnectorImpl"/>
```

El fitxer de configuració haurà d'incloure les dependències de Documentum que apareixen al propi fitxer de configuració de Documentum (project.xml) :

```
<dependencies>  
  <dependency>  
    <groupId>openFrame</groupId>  
    <artifactId>openFrame-services-exceptions</artifactId>  
    <version>${version.openframe}</version>  
    <type>jar</type>  
    <properties>  
      <war.bundle>true</war.bundle>
```



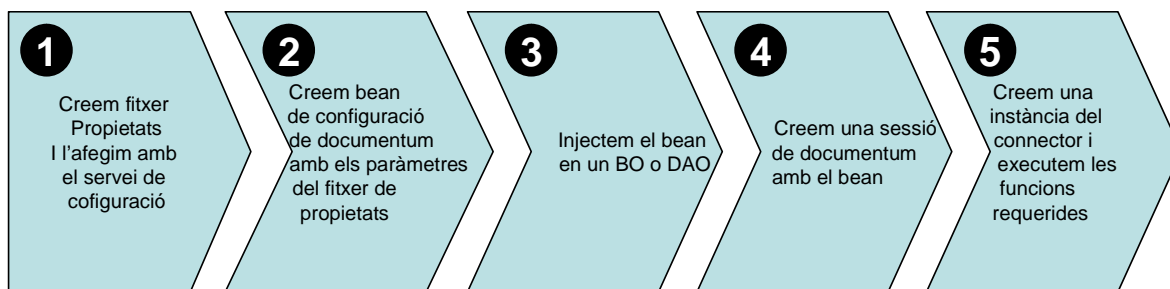
```
</dependency>
<dependencies>
  <groupId>openFrame</groupId>
  <artifactId>openFrame-services-logging</artifactId>
  <version>${version.openframe}</version>
  <type>jar</type>
  <properties>
    <war.bundle>true</war.bundle>
  </properties>
</dependency>
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>${version.log4j}</version>
  <type>jar</type>
  <properties>
    <jar.manifest.classpath>true</jar.manifest.classpath>
    <jar.bundle>true</jar.bundle>
    <war.bundle>true</war.bundle>
  </properties>
</dependency>
<dependency>
  <groupId>commons-logging</groupId>
  <artifactId>commons-logging</artifactId>
  <version>${version.commonslogging}</version>
  <type>jar</type>
  <url>http://jakarta.apache.org/commons</url>
  <properties>
    <jar.manifest.classpath>true</jar.manifest.classpath>
    <jar.bundle>true</jar.bundle>
    <war.bundle>true</war.bundle>
  </properties>
</dependency>
<dependency>
  <groupId>dfc</groupId>
  <artifactId>dfc</artifactId>
  <version>${version.dfc}</version>
  <type>jar</type>
  <properties>
    <jar.manifest.classpath>true</jar.manifest.classpath>
    <jar.bundle>true</jar.bundle>
    <war.bundle>true</war.bundle>
  </properties>
</dependency>
<dependency>
  <groupId>dfcbase</groupId>
  <artifactId>dfcbase</artifactId>
  <version>${version.dfcbase}</version>
  <type>jar</type>
  <properties>
    <jar.manifest.classpath>true</jar.manifest.classpath>
    <jar.bundle>true</jar.bundle>
    <war.bundle>true</war.bundle>
  </properties>
</dependency>
<dependency>
  <groupId>xml-apis</groupId>
  <artifactId>xml-apis</artifactId>
  <version>${version.xmlapis}</version>
  <type>jar</type>
  <properties>
    <jar.manifest.classpath>true</jar.manifest.classpath>
    <jar.bundle>true</jar.bundle>
    <war.bundle>true</war.bundle>
  </properties>
</dependency>
<dependency>
  <groupId>springframework</groupId>
```



```
<version>${version.spring}</version>
<artifactId>spring</artifactId>
<type>jar</type>
<url>http://www.springframework.org</url>
<properties>
  <jar.manifest.classpath>true</jar.manifest.classpath>
  <jar.bundle>true</jar.bundle>
  <war.bundle>true</war.bundle>
</properties>
</dependency>
</dependencies>
```

2.3. Utilització del Servei

La utilització del Connector es basa principalment en la configuració. L'ús directe des dels clients es permet mitjançant les interfícies definides.



2.3.1. Creació fitxer propietats

Hem de crear un fitxer de propietats (p.e. documentum.properties) amb els següents paràmetres:

```
documentum.user=usr
documentum.password=pwd
documentum.docBase=DTIDev
documentum.path=/CTTI
documentum.dir=dir CTTI
documentum.cabinet=CTTI
documentum.exportDir=c:\\Export_Dir
```

Definició dels paràmetres:

```
documentum.user=usuari del documentum
documentum.password=password documentum
documentum.docBase=doc base
documentum.path=path en documentum
documentum.dir=directori de documentum
```



```
documentum.cabinet=cabinet documentum  
documentum.exportDir=directori fitxers aplicació
```

Aquest fitxer de propietats l'inclourem al classpath de l'aplicació mitjançant el servei de configuració:

```
<beans>  
  <bean id="configurationService"  
    class="net.opentrends.openframe.services.configuration.springframework.beans.factory.config.HostPropertyPlaceholderConfigurer">  
    <property name="basePropertyFiles">  
      <list>  
        <value>classpath:documentum/documentum.properties</value>  
      </list>  
    </property>  
  </bean>  
</beans>
```

2.3.2. Creació bean d'Spring

Després haurem de crear un bean d'Spring de configuració de Documentum que inclogui aquestes variables. Aquest bean el crearem dins d'un fitxer xml dins del directori d'Spring (p.e. \spring\openFrame-connectors-documentum.xml):

```
<beans>  
  <bean id="documentumConfig"  
    class="net.gencat.pilot.connectors.documentum.DocumentumConfig">  
    <property name="user" value="${documentum.user}"/>  
    <property name="password" value="${documentum.password}"/>  
    <property name="docBase" value="${documentum.docBase}"/>  
    <property name="path" value="${documentum.path}"/>  
    <property name="cabinet" value="${documentum.cabinet}"/>  
    <property name="exportDir" value="${documentum.exportDir}"/>  
  </bean>  
</beans>
```

Podríem informar les propietats d'aquest bean directament, però és aconsellable fer-ho amb el fitxer de propietats, tal com acabem d'explicar.

2.3.3. Injecció del bean en el BO

En aquest bean tenim tota la informació necessària per connectar-nos a documentum. Podem injectar aquest bean, p.e., a un BO:



```
<bean lazy-init="true" id="documentBOTarget"
      class="net.gencat.pilot.model.bo.impl.DocumentBOImpl">
  <property name="documentumConfig" ref="documentumConfig" />
  <property name="documentDAO" ref="documentDaoTarget" />
</bean>
```

2.3.4. *Inici sessió documentum i ús*

En el BO ja podem accedir a Documentum:

```
public String openDocument(String idDocumentum) throws Exception {
    String exportDir = null;
    try {
        if (idDocumentum != null) {
            dS = DocumentumServiceImpl.getInstance();
            dci = new DocumentumConnectorImpl();

            sessio = dS.login(documentumConfig.getUser(),
                             documentumConfig.getPassword(),
                             documentumConfig.getDocBase(), dci);

            dci.exportarFitxer(documentumConfig.getExportDir(),
                              idDocumentum, sessio);

            exportDir = documentumConfig.getExportDir();
        }

        return exportDir;
    } catch (DocumentumException de) {
        throw de;
    } catch (Exception e) {
        throw e;
    }
}
```



2.4. Integració amb Altres Serveis

2.4.1. Integració amb el Servei de Internacionalització

En els fitxers de configuració es defineixen claus que permeten especificar quins missatges retornar en cas errors. Per a poder traduir aquestes claus és necessari especificar que el connector usará el Servei d'Internacionalització (veure Configuració).