



OpenText™ Documentum™ Content Management

Server System Object Reference Guide

Build a content or workgroup management application that accesses and uses repository objects.

EDCCS250400-ORD-EN-01

**OpenText™ Documentum™ Content Management
Server System Object Reference Guide**
EDCCS250400-ORD-EN-01
Rev.: 2025-Oct-17

This documentation has been created for OpenText™ Documentum™ Content Management CE 25.4.
It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product,
on an OpenText website, or by any other means.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111

Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440

Fax: +1-519-888-0677

Support: <https://support.opentext.com>

For more information, visit <https://www.opentext.com>

© 2025 Open Text

Patents may cover this product, see <https://www.opentext.com/patents>.

Disclaimer

No Warranties and Limitation of Liability

Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However,
Open Text Corporation and its affiliates accept no responsibility and offer no warranty whether expressed or implied, for the
accuracy of this publication.

Table of Contents

1	Introduction	11
2	Object basics	13
2.1	Computed properties	13
2.2	Property datatypes	25
2.3	Referencing properties	27
2.4	Identifiers	27
2.5	Relationships	30
3	Persistent object reference	31
3.1	Core persistent object types	31
3.2	ACL	31
3.3	Accelerated Content Services config	36
3.4	Activity	43
3.5	Activity group instance	61
3.6	Aggr domain	63
3.7	Alias set	64
3.8	App ref	66
3.9	Application	67
3.10	Aspect relation	70
3.11	Aspect type	71
3.12	Assembly	72
3.13	Atmos store	74
3.14	Attachments folder	74
3.15	Audit trail	75
3.16	Audit trail ACL	83
3.17	Audit trail Attrs	85
3.18	Audit trail group	86
3.19	Auth config	89
3.20	BLOB store	90
3.21	Branch Office Caching Services config	91
3.22	Builtin expr	96
3.23	Business Workspace Type Definition	97
3.24	Business Workspace template	99
3.25	Business Workspace instance	101
3.26	Business Workspace aspect	102
3.27	Ca store	103
3.28	Cabinet	107
3.29	Cache config	108
3.30	Category	110
3.31	Category assign	114

3.32	Category class	117
3.33	Change record	121
3.34	CI config	122
3.35	Class	125
3.36	Client registration	126
3.37	Client rights	127
3.38	Client rights domain	129
3.39	Completed workflow	131
3.40	Completed workitem	135
3.41	Component	140
3.42	Composite predicate	141
3.43	Cond expr	142
3.44	Cond ID expr	143
3.45	Config scope relation	144
3.46	Constraint set	145
3.47	Cont transfer config	146
3.48	Containment	149
3.49	Content	151
3.50	Cryptographic key	161
3.51	DD attr info	162
3.52	DD common info	168
3.53	DD info	174
3.54	DD type info	182
3.55	Display config	184
3.56	Distributed store	187
3.57	Messaging Service config	188
3.58	Docbase config	190
3.59	Docset	205
3.60	Docset run	206
3.61	Document	208
3.62	Domain	208
3.63	Dump object record	211
3.64	Dump record	212
3.65	Email message	216
3.66	Esign template	217
3.67	Expr code	218
3.68	Expression	219
3.69	External file store	220
3.70	External free store	221
3.71	External store	222
3.72	External URL store	224
3.73	Federation	224

3.74	File store	228
3.75	Folder	229
3.76	Foreign key	231
3.77	Format	231
3.78	FT engine config	235
3.79	FT filter config	237
3.80	FT index agent config	238
3.81	FT high water mark	242
3.82	dm_ftquery_subscription	244
3.83	Fulltext index	247
3.84	Func expr	249
3.85	Group	252
3.86	Index	258
3.87	Jar	260
3.88	Java library	260
3.89	JMS config	261
3.90	Job	265
3.91	Job request	271
3.92	Job sequence	272
3.93	Key	274
3.94	LDAP config	275
3.95	Link record - Deprecated	282
3.96	Linked store - Deprecated	283
3.97	Literal expr	285
3.98	Load object record	285
3.99	Load record	287
3.100	Location	289
3.101	Locator	291
3.102	Media profile	292
3.103	Message address	294
3.104	Message archive	294
3.105	Message attachment	296
3.106	Message queue config	296
3.107	Message container	297
3.108	Message route	299
3.109	Message route user data	299
3.110	Message user data	299
3.111	Metamodel	299
3.112	Method	300
3.113	MIP configuration	305
3.114	MIP trust configuration	306
3.115	Module	307

3.116	Module config	309
3.117	Mount point	310
3.118	Network location map	312
3.119	NLS DD info	314
3.120	Note	317
3.121	OTDS license config	317
3.122	Other file	319
3.123	Output device	319
3.124	Package	320
3.125	Partition scheme	324
3.126	Plugin	326
3.127	Policy	327
3.128	Preset info	339
3.129	Preset package	340
3.130	Procedure	341
3.131	Process	341
3.132	Process correlation set	349
3.133	Process parameter	350
3.134	Public key certificate	352
3.135	Qual comp	354
3.136	Query	356
3.137	Queue item	357
3.138	Reference	367
3.139	Registered	373
3.140	Registry	374
3.141	Relation	377
3.142	Relation SSA policy	380
3.143	Relation type	380
3.144	Relationship def	384
3.145	Replica record	389
3.146	REST store	391
3.147	Retainer	392
3.148	Routecase condition	398
3.149	S3 store	403
3.150	Scope config	404
3.151	Script	405
3.152	Sequence	406
3.153	Server config	407
3.154	Session	426
3.155	Smart list	426
3.156	SMTP config	427
3.157	SSA policy	429

3.158	State extension	430
3.159	State type	430
3.160	Store	431
3.161	Subcontent	441
3.162	SysObject	443
3.163	Sysprocess config	458
3.164	Taxonomy	458
3.165	TCF activity template	459
3.166	TCF activity	460
3.167	Transition condition	460
3.168	Type	464
3.169	Type info	469
3.170	User	473
3.171	Validation descriptor	488
3.172	Validation module	489
3.173	Validation relation	491
3.174	Value assist	492
3.175	Value func	494
3.176	Value list	494
3.177	Value query	495
3.178	ViPR store	497
3.179	Vstamp	497
3.180	WF attachment	498
3.181	WF package report parent	499
3.182	WF package schema	500
3.183	WF package skill	501
3.184	WF package type info	502
3.185	WF timer	503
3.186	Work item	505
3.187	Work queue	510
3.188	Work queue category	511
3.189	Work queue doc profile	512
3.190	Work queue policy	513
3.191	Work queue user profile	514
3.192	Workflow	515
3.193	Wq skill info	521
3.194	Wq task skill	523
3.195	Wq user skill	524
3.196	XML application	525
3.197	XML config	526
3.198	XML custom code	527
3.199	XML style sheet	528

3.200	XML zone	529
4	Lightweight object types and shareable object type properties	531
4.1	Lightweight and shareable objects	531
4.2	Lightweight	531
4.3	Lightweight, shareable, and non-qualifiable properties	532
5	Non-persistent object reference	535
5.1	Non-persistent object types	535
5.2	Client config	535
5.3	Connection config	535
5.4	Docbase locator	538
5.5	Docbroker locator	540
5.6	Server locator	541
5.7	Session config	543
6	Content Aviator object reference	551
6.1	dm csai config	551
6.2	dcis document aspect	552
6.3	dcis folder aspect	553
7	Collaboration object reference	555
7.1	Collaboration object types	555
7.2	Calendar	555
7.3	Calendar event	555
7.4	Comment	558
7.5	Datable	559
7.6	Datable row	560
7.7	Datable schema	560
7.8	Datable schema ex	562
7.9	Datable settings	562
7.10	Discussion	563
7.11	Notepage	564
7.12	Readcomment	564
7.13	Richtext	565
7.14	Room	567
7.15	Topic	569
8	Document Submission Manager object reference	571
8.1	Document Submission Manager object types	571
8.2	DSM application	571
8.3	DSM backbone	572
8.4	DSM doc properties	572

8.5	DSM drug product	574
8.6	DSM drug substance	574
8.7	DSM excipient	575
8.8	DSM facilities equip	576
8.9	DSM indication	576
8.10	DSM M1 backbone	577
8.11	DSM safety eval	578
8.12	DSM sect doc attributes	579
8.13	DSM section	580
8.14	DSM section extension	581
8.15	DSM stf backbone	582
8.16	DSM stf section	583
8.17	DSM study attributes	584
8.18	DSM study report	585
8.19	DSM submission	586
9	Retention Policy Services object reference	587
9.1	Retention Policy Services object types	587
9.2	Retention Policy Services action	587
9.3	Retention Policy Services action rel	588
9.4	Retention Policy Services authority	589
9.5	Retention Policy Services base date	590
9.6	Retention Policy Services child strategy	591
9.7	Retention Policy Services condition	593
9.8	Retention Policy Services contact	594
9.9	Retention Policy Services disposition method	595
9.10	Retention Policy Services event	597
9.11	Retention Policy Services execution rule	598
9.12	Retention Policy Services hold	599
9.13	Retention Policy Services notification	600
9.14	Retention Policy Services retainer	601
9.15	Retention Policy Services retainer event rel	604
9.16	Retention Policy Services retention policy	605
10	Interactive Delivery Services object reference	609
10.1	Interactive Delivery Services object types	609
10.2	SCS admin config	609
10.3	Webc config	610
10.4	Webc target	617
11	Forms Builder object reference	625
11.1	Forms Builder object types	625
11.2	XFM adaptor config	625

11.3	XFM form	627
11.4	XFM instance	628
11.5	XFM schema	629
12	Editorial Publishing Service type reference	631
12.1	Editorial Publishing Services types	631
12.2	EPS config cache	631
12.3	EPS elem type mapping	632
12.4	EPS element	632
12.5	EPS issue	634
12.6	EPS mapping	634
12.7	EPS metadata mapping	635
12.8	EPS page detail	636
12.9	EPS performer mapping	637
12.10	EPS placement	638
12.11	EPS pubinfo cache	639
12.12	EPS publication	639
12.13	EPS relation	640
12.14	EPS section	641
12.15	EPS state	642
12.16	EPS state mapping	643
12.17	EPS state perm mapping	644
12.18	EPS state transition	645
12.19	EPS template mapping	646
12.20	EPS transition override	646
12.21	EPS WF template mapping	647
12.22	EPS workflow data	648
A	Language and country codes	651
A.1	Language and country codes	651
B	RDBMS tables	655
B.1	RDBMS tables for OpenText Documentum CM types	655
C	Obsolete, deprecated, or unused objects	657
C.1	Obsolete, deprecated, or unused object types	657

Chapter 1

Introduction

This guide is for application developers, system administrators, and any readers who want to build a content or workgroup management application that accesses and uses repository objects. This guide assumes that you are familiar with the concepts of document processing, object-oriented programming, client-server applications, and working knowledge of SQL.

Chapter 2

Object basics

2.1 Computed properties

This section lists the computed properties supported by OpenText™ Documentum™ Content Management Server. Computed property values are not stored in the repository with an object's description but are computed at runtime when needed.

The “[Computed properties](#)” on page 13, describes all computed properties except those that are specific to lifecycles. The lifecycle-specific computed properties are described in the [computed lifecycle properties table](#).

Table 2-1: Computed properties

Property	Single or repeating	Description
_accessor_app_permit	R	Application_permit values of the entries for an ACL. The value at each index position represents the application_permit property value for the entry represented by the corresponding index position in _accessor_name.
_accessor_name	R	<p>The list of users and groups for whom some level of access to the selected object is defined. This property has valid values only if the repository is using ACLs for security.</p> <p>This property reflects the current status of an object's ACL. Therefore, if you are changing an object's permissions, this value reflects those changes, regardless of whether you have saved the changes. Check the _acl_ref_valid property to determine whether the values in _accessor_name have been saved.</p>

Property	Single or repeating	Description
_accessor_permit	R	<p>The permission levels assigned to each user or group defined in _accessor_name. The values, expressed as integers, are associated with the user or group at the corresponding index position. For example, the permission level at _accessor_permit[4] is assigned to the user or group specified by _accessor_name[4].</p> <p>This property reflects the current status of an object's ACL. Therefore, if you are changing an object's permissions, this value always reflects those changes, regardless of whether you have saved them. Check the _acl_ref_valid property to determine whether the values in _accessor_permit have been saved.</p>
_accessor_permit_type	R	<p>Permit types of the entries. The value at each index position is the permit type of the entry represented by the corresponding index position in _accessor_name.</p>
_accessor_xpermit	R	<p>Contains the integer value of the extended permissions assigned to each user or group returned by _accessor_name. The values, expressed as integers, are associated with the user or group at the corresponding index position. For example, the permission level at _accessor_xpermit[4] is assigned to the user or group specified by _accessor_name[4].</p>

Property	Single or repeating	Description
_accessor_xpermit_names	R	A list of the extended permissions, in string form, assigned to each user or group defined in _accessor_name.
_acl_ref_valid	S	A Boolean property that indicates whether the values of _accessor_name and _accessor_permit have been saved. If they have been saved, this property is FALSE. If the values have not been saved, _acl_ref_valid is TRUE.
_alias_set	S	The name of the alias set object bound to an object. This property is applicable to SysObjects, user objects, and server config objects.
_all_users_names	R	A list of all users directly or indirectly contained in a group. This property is cached on the server and client the first time it is queried in a session. Subsequent queries against it in the session return the cached values. To ensure that the returned values are correct, you can force the server to refresh the cache by issuing a IDfPersistentObject.revert method on the group object. Changing the group membership also invalidates the cache and forces a refresh.
_allow_change_location	S	A Boolean value indicating whether the user has the Change Location permission.
_allow_change_permit	S	A Boolean value indicating whether the user has the Change Permission permission.
_allow_change_state	S	A Boolean value indicating whether the user has the Change State permission.

Property	Single or repeating	Description
_allow_execute_proc	S	A Boolean value indicating whether the user has the Execute Procedure permission.
_allow_change_owner	S	A Boolean value indicating whether the user has the Change Ownership permission.
_change_folder_links	S	A Boolean value indicating whether the user has the Change Folder Links permission.
_attribute_list_values	R	<p>A string property containing the list of audited properties and their values recorded in an audit trail entry.</p> <p>The first index position stores the value of dm_audittrail.attribute_list. The remaining index positions stores the values from the dmi_audittrail_attrs object associated with the audittrail object, if any.</p> <p>The length of the property is the maximum length allowed by the underlying RDBMS for character data types.</p>
_cached	S	A Boolean property indicating whether the object is in the object cache. It returns TRUE if the object is in the cache and FALSE if not.
_changed	S	<p>A Boolean property that indicates whether the object has been changed since it was last saved.</p> <p>The server checks the object cache and if the object is not in the cache or is in the cache but hasn't been changed, server returns FALSE. TRUE is returned if the object is present in the cache and has been changed.</p>

Property	Single or repeating	Description
_componentID	R	<p>The chronicle IDs of the component documents that make up a virtual document. The ID value at each index position is the chronicle ID for the component at that position in the virtual document.</p> <p>The zero position, <code>_containID[0]</code>, represents the chronicle ID for the virtual document. A virtual document is always considered a component of itself.</p>
_containID	R	<p>The object IDs of the containment objects that link the component document to its containing virtual document. The ID value at each index position represents the containment object for the component at that position in the virtual document.</p> <p>The zero position, <code>_containID[0]</code>, represents the containment object for the virtual document. A virtual document is always considered a component of itself.</p>

Property	Single or repeating	Description
_content_state	R	<p>An integer value representing the state of a content for a given SysObject. State values and their meanings are:</p> <ul style="list-style-type: none"> • 0, meaning available and online • 1, meaning archived • 2, meaning offline • 3, meaning error fetching content from the repository <p>For example, if _content_state[1] = 1, then the first content for the object is archived.</p>
_count	S	The number of properties in the object.
_current_state	S	The current lifecycle state of the object.
_docbase_id	S	That portion of a specified object ID that corresponds to the repository ID.
_dump	S	A description of the object, including all properties and their values.
_has_config_audit	S	A Boolean property indicating whether the user has the privilege. T (TRUE) means the user has the privilege.
_has_create_type	S	A Boolean property indicating whether the user has the privilege. T (TRUE) means the user has the privilege.
_has_create_group	S	A Boolean property indicating whether the user has the privilege. T (TRUE) means the user has the privilege.

Property	Single or repeating	Description
_has_create_cabinet	S	A Boolean property indicating whether the user has the privilege. T (TRUE) means the user has the privilege.
_has_purge_audit	S	A Boolean property indicating whether the user has the privilege. T (TRUE) means the user has the privilege.
_has_superuser	S	A Boolean property indicating whether the user has the privilege. T (TRUE) means the user has the privilege.
_has_sysadmin	S	A Boolean property indicating whether the user has the privilege. T (TRUE) means the user has the privilege.
_has_view_audit	S	A Boolean property indicating whether the user has the privilege. T (TRUE) means the user has the privilege.
_id	S	A unique object identifier that contains the object's r_object_id value.
_is_restricted_session	S	A Boolean property that indicates whether the current session is a restricted session in which the only valid operation is changing the user's password. T means that the current session is such a session. F means that the session is not restricted to a change password operation.
_isdeadlocked	S	A Boolean property that indicates whether the current session was chosen as the victim of a deadlock. This property is reset when the next method that issues an RPC call is executed.

Property	Single or repeating	Description
_isnew	S	A Boolean property that indicates whether the object is a new object, that is, whether the object has been saved in the repository.
_isreplica	S	For use in a distributed storage environment, this indicates whether an object is a replica (read-only copy) or the original object. This property is valid only for cabinets, folders, documents, notes, and smart lists.
_istransactionopen	S	A Boolean property that indicates whether an explicit transaction is open. TRUE means a transaction is open. FALSE means there is no transaction open.
_lengths	R	Lengths of an object's properties. Property length is measured in the number of bytes required by its datatype, such as a 4-byte integer. However, this information is only useful for character string columns.
_masterdocbase	S	The repository ID of the repository that owns an object. Unless the object is a replica, this value is the ID of the local repository. If the object is a replica, the value is the ID of the foreign repository that owns the object. This property is an integer property.
_names	R	Names of an object's properties.

Property	Single or repeating	Description
_permit	S	<p>An integer number that corresponds to the permissions for an object of a specified user. If the user is unspecified in the query (for example, <code>get,c,object_id,_permit</code>), the current user is assumed.</p> <p>The permission returned represents the most permissive permission the user has for the specified object.</p> <p>A value of 0 indicates that the object is not a <code>SysObject</code>. Values of 2 to 7 correspond to the object-level permissions <code>Browse</code> through <code>Delete</code>, respectively.</p> <p>For more information about the permissions defined for objects, see <i>OpenText Documentum Content Management - Administrator User Guide (EDCAC250400-UGD)</i>.</p>
_policy_name	S	Name of the lifecycle to which the object is attached.
_repeating	R	Whether properties are single or repeating. Values are 0 for single-valued properties and 1 for repeating properties.
_resume_state	S	Name of the lifecycle's normal state which immediately preceded the current state of the object.
_sign_data	S	<p>An XML file that contains the properties used to generate the hashed signature for an audit trail entry.</p> <p>The output is well-formed XML. It includes the values of the entry's associated <code>dmi_audittrail_attrs</code> object, if one exists.</p>

Property	Single or repeating	Description
_status	S	<p>Level of the error (if any) resulting from a method call against an object. The level is represented as a numeric value. The values are:</p> <ul style="list-style-type: none"> • 0, for informational messages (OK, for example) • 1, for trace messages • 2, for warning messages • 3, for error messages • 4, for fatal error messages <p>If two or more errors occur, only the first error's level is stored unless the subsequent error is more severe. In such cases, the number associated with the more severe error is stored. If no errors occur, this property contains 0 (for OK). To clear this property, use the reset method.</p>
_type_id	S	The object ID of the dm_type object for the object type of the specified object. This is returned as a string value.
_type_name	S	The internal name of the object's type.
_types	R	Datatypes of an object's properties. The datatypes are specified as numbers: <ul style="list-style-type: none"> • 0 for Boolean • 1 for Integer • 2 for String • 3 for ID • 4 for Time • 5 for Double
_typestring	S	Returns the number of supertypes in a specified object's hierarchy and type information about each supertype.

Property	Single or repeating	Description
_values	R	Total number of values of an property (useful for repeating properties).
_xpermit	R	Contains the integer value of the extended permissions the current user or the specified user has on the object.
_xpermit_list	S	Contains a full list of the extended permissions, in string form, currently supported by server. This computed property returns the same list regardless the object ID.
_xpermit_names	S	Contains the list of the extended permissions, in string form, the current user or the specified user has on the object.

The [computed lifecycle properties table](#), describes additional computed properties that apply only to lifecycles (dm_policy objects).

Table 2-2: Computed lifecycles properties

Property	Single or repeating	Description
_alias_sets	R	List of the object names of the valid alias sets for the lifecycle. The value at each index position corresponds to the alias set specified at the same index position in the alias_set_ids property.

Property	Single or repeating	Description
_entry_criteria	R	<p>List of expressions that must evaluate to TRUE before an object can be promoted to a state. The value at each index position corresponds to the conditions set for the state defined at that index position.</p> <p>This computed property is read and write. Setting _entry_criteria sets the entry_criteria_id property in the policy object. For more information about the entry_criteria_id property, see <i>OpenText Documentum Content Management - Server Fundamentals Guide</i> (EDCCS250400-GGD).</p>
_included_types	S	The list of acceptable object types for this policy, in lowercase, separated by commas.
_next_state	R	The name of next state. Exception states and the terminal state do not have a next state.
_previous_state	R	The name of previous state. Exception states and the base state do not have a previous state.
_state_extension_obj[[index]]	R	<p>The object ID of the state extension object associated with a particular lifecycle state.</p> <p>The index value identifies the lifecycle state by identifying the state's state number. For example, _state_extension_obj[0] returns the object ID of the state extension object associated with the lifecycle state number 0.</p> <p>If an index value is not included, the default is 0.</p>

Property	Single or repeating	Description
_state_type	R	<p>Identifies whether the state is an exception state or, if is not an exception state, it's place in the sequence of lifecycle states. The information at each index position is applicable to the state whose state number corresponds to the index position number. Valid values are:</p> <ul style="list-style-type: none"> • 1, for Exception <ul style="list-style-type: none"> – 0, for Base (first normal state) – 1, for Terminal (last normal state) • 2, for Step (normal between base and terminal states)

2.2 Property datatypes

The [property datatypes table](#), describes the OpenText™ Documentum™ Content Management datatypes, their RDBMS equivalents, their range of values, and their default values. The length specifications for the string datatype are expressed as number of bytes.

Table 2-3: Property datatypes

Property datatypes	DQL type	Oracle	Microsoft SQL Server	PostgreSQL	Value range	Default value
DM_INTEGER	integer	number(10)	integer	integer	-214748364 8 to +214748364 7	0
DM_BOOLEAN	boolean, bool	number(6)	integer	integer	1 (TRUE) or 0 (FALSE)	0 (FALSE)

Property datatypes	DQL type	Oracle	Microsoft SQL Server	PostgreSQL	Value range	Default value
DM_STRING	string, character, char	char(64) max length is 4000 [1]	nvarchar and nchar on SQL Server, with maximum length of 4000 (Unicode) or 7000 (non-Unicode)	character varying	0 (blank string) to maximum for RDBMS (PostgreSQL) maximum length: 10485760 or in any case, the longest possible. Character string that can be stored is about 1 GB.	a null string, specified ""
DM_DOUBLE	double, float	number 1 x 10-129 to 1 x 10+129	float 1.7E-308 to 1.7E+308	double precision	specific to RDBMS (PostgreSQL) 15 decimal digits precision	0.0
DM_TIME	time, date	date	datetime (MS SQL Server)	timestamp without timezone	01/01/1753 to 12/31/9999	current time
DM_ID	ID	char(16)	char(16)	char(16)	not applicable	an ID of all zeros

[1] The maximum length of a non-qualifiable property defined as a string datatype is the value in the `max_nqa_string` key in the `server.ini` file. If not set, the maximum length of a non-qualifiable string property is the maximum length defined by the underlying relational database.



Note: The lengths defined for a string-datatype property and in the reference descriptions of object types are expressed as bytes. For example, the `dm_sysobject.title` property is `string(255)`. This reference means that it is a string datatype that accepts a value that is a maximum of 255 bytes long.

2.3 Referencing properties

You can reference persistent properties in OpenText™ Documentum™ Content Management Foundation Java API method calls or DQL statements. Computed properties can be referenced only in OpenText Documentum Content Management (CM) Foundation Java API calls; you cannot reference them in DQL statement.

Foundation Java API provides get and set methods for most properties. For those that do not have specific get and set methods, you can use the get *Datatype* and set *Datatype* methods in the IDfTypedObject interface. For example, if you were retrieving the value of a string property, you could call IDfTypedObject.getString with the property name as the argument. There is a get and set method for each supported datatype in the IDfTypedObject interface.

When you reference a property in DQL, use the property's name. For example:

```
1 SELECT "title", "subject" FROM "dm_document"
2 WHERE ANY "keywords" IN ('floorplan', 'cubicle')
```

If the property is a property defined for an aspect, qualify the property name with the aspect name. For example:

```
1 SELECT "title", "editingAspect.editor_name" FROM "dm_document"
2 WHERE ANY "keywords" IN ('floorplan', 'cubicle')
```

If the name matches a DQL reserved word, you must double-quote the name when you reference it in a method call or DQL statement.

2.4 Identifiers

This section describes the identifiers that OpenText Documentum Content Management (CM) Server recognizes.

2.4.1 Names

Names identify types, properties, tables, columns, users, and groups.

Names fall into three groups, depending on which naming rules apply to them. The groups are:

- Object type and property names
- User and group names
- Table and column names

Three naming rules apply to all names:

- Do not begin a name with the prefix dm. OpenText Documentum Content Management (CM) reserves this prefix for its own use.
- The name must consist of ASCII characters.

- If a name matches a DQL reserved word, enclose the name in double quotes whenever you use it in a DQL query.



Note: It is good practice to choose names that do not conflict with DQL reserved words.

In addition to these rules, other rules apply specifically to each group of names.

2.4.1.1 Object type and property names

Object type and property names must observe the following rules:

- The maximum allowed length is 27 characters.
- The first character must be a letter. The remaining characters can be letters, digits, or underscores.
- The name cannot contain spaces or punctuation.
- The name cannot be any of the words reserved by the underlying RDBMS.

In addition, object type names cannot end in an underscore (_).

Types are not case-sensitive, but property names must be in lowercase in DQL and Foundation Java API.

2.4.1.2 User and group names

User and group names must observe the following rules:

- The maximum allowed length is 255 characters.
- If the name is enclosed in single quotes when referenced, it can contain any character allowed in a character string literal (spaces, apostrophes, and so on).
- If the name is enclosed in double quotes when referenced, it can be a DQL reserved word.

User names are case sensitive. Group names are not case sensitive. All group names are stored in lowercase in the repository regardless of the case used for the name when creating the group.

2.4.1.3 Table and column names

Table and column names must observe the following rules:

- The maximum length allowed by Documentum CM Server is 64 characters. The RDBMS or operating system may impose stricter limits on the name's length.
- The first character must be a letter. The remaining characters can be letters, digits, or underscores.
- The name cannot contain spaces or punctuation.
- The name cannot be any of the words reserved by the underlying RDBMS.

Table and column names may or may not be case sensitive, depending on the rules imposed by the RDBMS or operating system.

2.4.2 Object identifiers

Object identifiers, or object IDs, are assigned by Documentum CM Server when an object is created. You cannot assign them yourself nor can you change them. Object IDs are expressed as a 16-character string.

2.4.3 Property identifiers

Property identifiers are assigned by Documentum CM Server when the property is defined for an object type. That can occur when the object type is created or when a type is altered to add a property.

Property identifiers are unique within the object type's type hierarchy. The identifiers cannot be changed. They are used instead of property names, to identify properties when properties are stored in a property bag.

2.4.4 Collection identifiers

A collection identifier represents the non-persistent object (collection) that contains the results of a query or a method call. When the server executes the query or method that returns a collection, it places the results in a collection and returns an identifier for the collection to the user or application. The user or application must use that identifier to retrieve the results.

Collection identifiers are two characters. The first is the letter q and the second is a numeric character.

2.4.5 Type identifiers

A type identifier is a two-character string that appears as the first two characters of an object's object ID. Each system-defined object type has a unique type identifier. For example, the type identifier for the dm_document type is 09, and all objects of type dm_document have an object ID that begins with 09.

When you create an object type, it inherits its type identifier from the system-defined supertype. For example, any user-defined subtypes of dm_document also have a type identifier of 09.

User-defined types that have no supertype have a type identifier of 00.

2.4.6 Session identifiers

Session identifiers are used in trace file entries to identify the session associated with an entry. A session identifier has the formats *n*, where *n* is an integer number.

2.5 Relationships

A relationship is an association between two objects in the repository. Some system-defined relationships are installed with Documentum CM Server. For example, annotations are implemented as a system-defined relationship between a SysObject, generally a document, and a note object. Documentum CM Server also allows defining custom relationships.

Each kind of relationship is described by a dm_relation_type object. The properties of a relation type object name the relationship, define the security that applies to instances of that relationship, and define the behavior when an object participating in an instance of that relationship is copied, versioned, or deleted.

Each instance of a particular relationship is described by a dm_relation object. A relation object identifies the two objects involved in the relationship and describes the type of relationship. Relation objects also have several properties that are used to manage and manipulate the relationship.

Both types, dm_relation_type and dm_relation, are persistent. Both object types have object IDs and are stored in the repository. However, relation type or relation objects cannot be versioned, nor are object-level permissions applied to these object types. Security for relationships is set by the value defined for the security_type property of the relation type object.

Chapter 3

Persistent object reference

3.1 Core persistent object types

Core persistent object types are either built in to Documentum CM Server or installed through a script when a repository is configured. Most of the types are direct or indirect subtypes of the persistent object type. The persistent object type has three properties that it passes to all of its subtypes:

- `r_object_id`

The `r_object_id` property contains a 16-character hexadecimal string that is assigned by the system when an object is created. This value uniquely identifies the object in the repository.

- `i_vstamp`

The `i_vstamp` property contains an integer value that represents the number of committed transactions that have changed an object. This value is used for versioning, as part of the locking mechanism, to ensure that one user does not overwrite the changes made by another.

- `i_is_replica`

The `i_is_replica` property indicates whether the object is a local replica of an object in a remote repository.

The non-persistent types described in this chapter are not stored in the repository. Consequently, they do not have object IDs or `i_vstamp` values, and they are never replicated. They are objects that are created and used during a session and vanish when the session is terminated.

This chapter lists the properties defined for the persistent and non-persistent types in Documentum CM Server. For the persistent types, the information also includes the type's supertype, subtypes, and internal name.

3.2 ACL

An ACL object represents an Access Control List.

3.2.1 Description

- Supertype: Persistent
- Object Subtypes: None
- Internal name: dm_acl
- Object type tag: 45

For SysObjects, the entries in the repeating properties control who can access the object to which the ACL is attached. If the repository security mode is set to acl, Documentum CM Server enforces the permissions granted by the ACL entries.

3.2.2 Properties

The table describes the object properties.

Table 3-1: ACL type properties

Property	Datatype	Single or repeating	Description
acl_class	integer	S	Specifies whether the ACL is a regular ACL, a template, an instance of a template, or a public ACL. Valid values are: <ul style="list-style-type: none">• 0: Regular ACL• 1: Template ACL• 2: Template instance• 3: Public ACL
description	string(128)	S	User-defined description of the ACL.
globally_managed	Boolean	S	Specifies whether the acl object is managed globally or locally. The default is FALSE, meaning that it is locally managed.
i_has_access_restrictions	Boolean	S	Specifies whether the ACL has one or more entries with a permit_type value of 2.

Property	Datatype	Single or repeating	Description
i_has_required_groups	Boolean	S	Specifies whether the ACL has one or more entries with a permit_type value of 3.
i_has_required_group_set	Boolean	S	Specifies whether the ACL has one or more entries with a permit_type value of 4.
i_partition	integer	S	Specifies the partition where the object resides.
object_name	string(32)	S	Names the ACL object. The name, if provided, must be unique among the ACLs in the repository. This property can be NULL.
owner_name	string(255)	S	Specifies the owner of the ACL. This will be the user who created the ACL or, for system-level ACLs, the name of the repository owner or the alias dm_dbo.
r_accessor_name	string(255)	R	<p>List of users that have access to the object attached to the ACL. Valid entries in the list are individual user names, group names, and aliases.</p> <p>This must be or resolve to a group name if the r_permit_type is RequiredGroup or RequiredGroupSet.</p>

Property	Datatype	Single or repeating	Description
r_accessor_permit	integer	R	<p>Specifies the access level granted to the user or group at the corresponding index level. Valid values are:</p> <ul style="list-style-type: none">• 0: Null• 1: None• 2: Browse• 3: Read• 4: Relate• 5: Version• 6: Write• 7: Delete <p>For more information about the access level, see <i>OpenText Documentum Content Management - Administrator User Guide</i> (EDCAC250400-UGD).</p>
r_accessor_xpermit	integer	R	<p>Specifies the extended permission level granted to the user or group at the corresponding index level.</p> <p>Extended permissions are:</p> <ul style="list-style-type: none">• execute_proc• change_location• change_state• change_permit• change_owner• delete_object• change_folder_links

Property	Datatype	Single or repeating	Description
r_application_permit	string(128)	R	<p>Specifies a permission recognized by an application. An application permit is not recognized or enforced by Documentum CM Server. The application is responsible for enforcing the permission.</p> <p>The permission defined at a particular index position is applicable to the user or group identified in the corresponding position in r_accessor_name.</p>
r_is_group	Boolean	R	Specifies whether the r_accessor_names at the corresponding index levels are individual users or groups.
r_has_events	Boolean	S	Specifies whether someone has registered the ACL for auditing.
r_is_internal	Boolean	S	Indicates whether the ACL was created explicitly by a user or implicitly by the server.

Property	Datatype	Single or repeating	Description
r_permit_type	integer	R	<p>Specifies the permit type for the entry. Valid values are:</p> <ul style="list-style-type: none">• 0: AccessPermit• 1: ExtendedPermit• 2: ApplicationPermit• 3: AccessRestriction• 4: ExtendedRestriction• 5: ApplicationRestriction• 6: RequiredGroup• 7: RequiredGroupSet <p>To set this value to either 6 or 7, the value in r_accessor_name at the corresponding index position must be a group name.</p> <p>The default value is 0.</p>

3.3 Accelerated Content Services config

Stores the configuration information for an OpenText™ Documentum™ Content Management Accelerated Content Services server.

3.3.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_acs_config
- Object type tag: 08

An acs config object describes the configuration of an Accelerated Content Services server. One Accelerated Content Services server is installed with each Documentum CM Server. The installation automatically creates the associated acs config object. To change or modify the settings, use Documentum Administrator or IAPI.

3.3.2 Properties

The table describes the object properties.

Table 3-2: Accelerated Content Services config type properties

Property	Datatype	Single or repeating	Description
acs_base_url	string(240)	R	The base URL to use when accessing the Accelerated Content Services using the protocol identified at the corresponding index position in acs_supported_proto_col.
acs_network_locations	string(80)	R	The network identifiers of the network locations served by the Accelerated Content Services.

Property	Datatype	Single or repeating	Description
acs_rw_capability	integer	S	<p>Specifies whether the Accelerated Content Services is enabled and whether it can use surrogate get. Valid values are:</p> <ul style="list-style-type: none"> • 0: Accelerated Content Services is disabled • 1: Accelerated Content Services is enabled • 3: SurrogateGet is enabled
acs_supported_protocol	string(6)	R	<p>The set of network protocols supported by the Accelerated Content Services.</p> <p>Values in this property must be all lowercase. For example, http or https.</p>
config_type	integer	S	<p>Specifies whether the accessible storage areas and defined projection targets are configured in the server config object or this acs config object. Valid values are:</p> <ul style="list-style-type: none"> • 1: Server config object (identified in svr_config_id) • 2: Acs config object (defined in the near_stores and projection-related properties)

Property	Datatype	Single or repeating	Description
delivery_type	string(4)	S	<p>Specifies whether the OpenText™ Documentum™ Content Management Messaging Service server can send messages to the Accelerated Content Services server or must wait for the Accelerated Content Services server to pull messages from the OpenText Documentum Content Management (CM) Messaging Service server. Valid values are:</p> <ul style="list-style-type: none">• push: The Messaging Service can send messages to this server.• pull: The Messaging Service must wait for the server to pull messages. <p>The default value is push.</p>

Property	Datatype	Single or repeating	Description
dormancy_status	string(32)	S	<p>Specifies whether the Accelerated Content Services server is active or dormant.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • ACTIVE: The Accelerated Content Services server is active. • DORMANT: The Accelerated Content Services server is dormant. <p>When the Accelerated Content Services server is restarted or re-initialized, its dormant status is set to the same value as this attribute.</p>
is_cache_acs	Boolean	S	<p>The value is F (FALSE) for Accelerated Content Services servers.</p> <p>The T (TRUE) value is not supported for Accelerated Content Services servers.</p>
near_stores	string(32)	R	<p>Storage areas that are accessible for this Accelerated Content Services server.</p> <p>The default is a single blank.</p>
projection_enable	Boolean	R	Specifies whether projection to the connection broker specified at the corresponding index position in projection_targets is enabled.

Property	Datatype	Single or repeating	Description
projection_netloc_enable	Boolean	R	Specifies whether the server is allowed to service users in the network location specified at the corresponding index position in projection_netloc_iden.
projection_netloc_iden	string(80)	R	Names of the network locations that this server can service. The corresponding index position in projection_netloc_enable must be set to T (TRUE) to allow the server to actually handle requests from users in a particular network location specified in this property.
projection_ports	integer	R	Specifies the port on which the connection broker is listening. The value at each index position is matched to the connection broker specified at the corresponding level in projection_targets. The default port is 1489.
projection_proxval	integer	R	Proximity values representing the Accelerated Content Services server's proximity to the network locations identified at the corresponding index position in projection_netloc_iden.

Property	Datatype	Single or repeating	Description
projection_targets	string(80)	R	Names of the host machines on which the connection brokers reside.
server_major_version	string(16)	S	Used internally.
server_minor_version	string(16)	S	Used internally.
supported_compression_mode	integer	R	Specifies whether the Accelerated Content Services supports compression for the storage type identified in the corresponding index position of supported_store_types. Valid values are: <ul style="list-style-type: none"> • 0: Compression is not supported. • 1: Compression is supported.
supported_crypto_mode	integer	R	Specifies whether the Accelerated Content Services supports encryption for the storage type identified in the corresponding index position of supported_store_types. Valid values are: <ul style="list-style-type: none"> • 0: Encryption is not supported • 1: Encryption is supported

Property	Datatype	Single or repeating	Description
supported_store_types	integer	R	<p>Set of storage area types supported by the Accelerated Content Services. Valid values are:</p> <ul style="list-style-type: none"> • 1: File store • 3: Linked store. Linked storage areas are deprecated in OpenText Documentum CM 6.0. • 4: Distributed store • 5: Blob store • 7: External store • 8: External file store • 9: External URL store • 10: External free store • 11: Ca store
svr_config_id	ID	S	Object ID of the server config object of the Accelerated Content Services server's associated Documentum CM Server.

3.4 Activity

An activity object defines a workflow activity.

3.4.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_activity
- Object type tag: 4c

The properties in an activity object define who performs the activity and the packages and work items generated from the activity.

3.4.2 Properties

The table describes the object properties.

Table 3-3: Activity type properties

Property	Datatype	Single or repeating	Description
activity_group_flag	integer	S	Reserved for future use.
activity_group_id	ID	S	Reserved for future use.
activity_urn	string (512)	S	Activity uniform resource name.
control_flag	integer	S	Controls who receives a task if automatic delegation fails. Valid values are: <ul style="list-style-type: none">• 0: The task is reassigned to the workflow supervisor.• 1: The task is reassigned to the original task performer.

Property	Datatype	Single or repeating	Description
exec_err_handling	integer	S	<p>Specifies how to handle execution failure for a runtime instance of this activity. Valid values are:</p> <ul style="list-style-type: none"> • 0: Stop if the activity fails. • 1: Proceed with the transition even if the activity fails. • 2: Terminate the workflow. <p>The default is 0.</p>
exec_method_id	ID	S	Contains the ID of the dm_method object that represents the application. This is required if exec_type is automatic.
exec_retry_interval	integer	S	<p>The length of time, in minutes, to wait between each attempt to execute an automatic activity.</p> <p>The default is 0, meaning retry immediately.</p>
exec_retry_max	integer	S	<p>The maximum number of times to attempt to execute an automatic activity before taking the failure action defined in the exec_err_handling property.</p> <p>The default is 0, meaning do not make any execution retries.</p>

Property	Datatype	Single or repeating	Description
exec_save_results	Boolean	S	<p>Specifies whether to save results generated by the application. The default is F (FALSE).</p> <p>If set to T (TRUE), the server does not verify the execution results and assumes that the method executed properly. Execution results are saved for later verification in the object whose ID is recorded in the r_exec_result_id property of the work item.</p>
exec_subtype	integer	S	<p>Indicates whether the activity waits for an inbound message to proceed at runtime. Valid values are:</p> <ul style="list-style-type: none"> • 0: Begin, Step or End activity and no inbound message is required to start the activity. • 1: Not an Initiate activity but it triggered by an inbound message. • 2: Initiate activity that is triggered by an inbound message. • 3: Initiate activity but it is not triggered by an inbound message. Typically, these activities have forms associated with them.

Property	Datatype	Single or repeating	Description
exec_time_out	integer	S	<p>Specifies how long an automatic activity's method is allowed to execute before timing out.</p> <p>This value must be set to a positive number if the activity is an automatic activity (exec_type = 1). The value is interpreted in seconds.</p> <p>This value is not used for manual activities.</p>
exec_type	integer	S	<p>Specifies the execution type. Valid values are:</p> <ul style="list-style-type: none"> • 0: Manual • 1: Automatic
is_private	Boolean	S	Specifies whether the definition is private or public. The default is F (private).
lsm_id	ID	S	LSM ID.
performer_flag	integer	S	<p>Specifies whether delegation or extension is allowed. Valid values are:</p> <ul style="list-style-type: none"> • 0: No delegation or extension • 1: Allow delegation. • 2: Allow extension. • 3: Allow both, delegation and extension. <p>The default is 0.</p>

Property	Datatype	Single or repeating	Description
performer_name	string(255)	R	<p>Specifies the activity performer. Valid values are a user name, a group name, the string dm_world, representing all users in the repository, a work queue name, or an alias. The alias format is</p> <pre>%[alias_set_name.]alias_name</pre> <p>The server always interprets names starting with a % as an alias, including user and group names.</p>
performer_type	integer	S	<p>Identifies the category of the activity performer. Valid values are:</p> <ul style="list-style-type: none"> • 0: Workflow supervisor. • 1: Repository owner. • 2: Last performer. • 3: A user. • 4: All members in a group. • 5: Any user in a group. • 6: The group member who is loaded last. • 8: Some members of a group or some users in the repository. • 9: Some members of a group or some users in the repository sequentially. • 10: A user from a work queue.

Property	Datatype	Single or repeating	Description
post_timer	integer	S	<p>Specifies the length of time, in hours, in which the activity must be completed after it is started.</p> <p>The default value is 0, which means the timer is not in use.</p>
post_timer_action	ID	R	Object IDs of the module config objects that identify the actions associated with the timer's firing. The ID at a particular index position represent the action associated with the triggering associated with the corresponding index position in post_timer_increment.
post_timer_calendar_flag	integer	S	Reserved for future use.
post_timer_calendar_id	ID	S	Reserved for future use.

Property	Datatype	Single or repeating	Description
post_timer_incremen t	integer	R	<p>Value in the first index position defines a time interval after which the timer is triggered if the activity has not been completed. The interval is expressed in minutes and is counted from the start of the activity.</p> <p>Values in the following index positions define intervals for second and subsequent triggering of the timer if the activity has not been completed. The intervals in these index positions are counted from the end of the previous interval.</p>
post_timer_name	string(64)	S	Contains names of the post-timers defined for this activity.
post_timer_repeat_la st	Boolean	S	Specifies whether to repeat the last action specified in the post_timer_action property.
pre_timer	integer	S	<p>Defines the length of time, in hours, in which the activity must be started after the workflow starts.</p> <p>The default value is 0, which means the timer is not in use.</p>

Property	Datatype	Single or repeating	Description
pre_timer_action	ID	R	Object IDs of the module config objects that identify the actions associated with the timer's firing. The ID at a particular index position represent the action associated with the triggering associated with the corresponding index position in pre_timer_increment.
pre_timer_calendar_flag	integer	S	Reserved for future use.
pre_timer_calendar_id	ID	S	Reserved for future use.
pre_timer_increment	integer	R	<p>Value in the first index position defines a time interval after which the timer is triggered if the activity has not been started. The interval is expressed in minutes and is counted from the start of the workflow.</p> <p>Values in the following index positions define intervals for second and subsequent triggerings of the timer if the activity has not been started. The intervals in these index positions are counted from the end of the previous interval.</p>
pre_timer_name	string(64)	S	Contains names of the pre-timers defined for this activity.

Property	Datatype	Single or repeating	Description
pre_timer_repeat_last	Boolean	S	Specifies whether to repeat the last action specified in the pre_timer_action property.
property_name	string(256)	R	Contains names of extra properties of the activity.
property_value	string(2000)	R	Value of the extra property defined at the corresponding index of property_name.
r_condition_id	ID	S	Contains the object ID of a compound condition (a dm_cond_expr object).
r_condition_name	string(16)	R	Contains the names of route cases.
r_condition_port	string(16)	R	Contains the output port names. The ports at a particular index position are the ports associated with the route case condition at the corresponding index position in r_condition_name.
r_definition_state	integer	S	Identifies the state of this activity definition. Valid values are: <ul style="list-style-type: none">• 0: Draft• 1: Validated• 2: Installed

Property	Datatype	Single or repeating	Description
r_package_flag	integer	R	<p>Indicates whether the package is visible and whether it is allowed to be empty. Valid values are:</p> <ul style="list-style-type: none"> • 0: The package is invisible but cannot be empty. • 1: The package is visible and cannot be empty. • 2: The package is invisible but may be empty. • 3: The package is visible and may be empty. <p>The default is 1.</p>
r_package_id	ID	R	Contains the ID of a single package component. Not used if the package has more than one component.
r_package_label	string(32)	R	Contains a version label for the package.
r_package_name	string(128)	R	Contains the package name. The package name must be unique within the scope of the containing port.
r_package_oprtn	string(64)	R	Contains the operation to be performed by performers.
r_package_report_flag	integer	R	Specifies whether to report the package defined at the corresponding index position of the r_package_name.
r_package_type	string(40)	R	Contains the type name of the package object.

Property	Datatype	Single or repeating	Description
r_performer_cond_id	ID	S	Object ID of dm_cond_expr object that records the conditions used to resolve the performer.
r_performer_cond_name	string(16)	R	Names of the conditions used to choose a performer.
r_performer_cond_user	string(2000)	R	Contains a user or group name, an alias that resolves to a user or group name, or a select query that returns a user or group name. The value at a particular index position applies to the condition specified at the corresponding index position in r_performer_condition_name.
r_port_name	string(16)	R	Contains the unique port identifier.
r_port_type	string(16)	R	Specifies the port type. Valid values are INPUT, OUTPUT, REVERT, or EXCEPTION.
r_predicate_id	ID	R	Object ID of a persistent predicate, to be evaluated for the corresponding route case.
repeatable_invoke	Boolean	S	Indicates whether this activity can be included multiple times or only once within a process definition. The default value is T (TRUE), meaning the activity may be included multiple times.

Property	Datatype	Single or repeating	Description
resolve_pkg_name	string(128)	S	Contains the name of the package whose associated alias set will be used to resolve the activity's performer alias. This property is used only if resolve_type is set to 1. If resolve_type is 1 and resolve_pkg_name is not set, then all incoming packages are checked in the order in which they are defined.
resolve_type	integer	S	Defines the resolution algorithm for the performer aliases. Valid values are: <ul style="list-style-type: none"> • 0: Normal resolution path (default). • 1: Alias set associated with incoming package. • 2: Alias set associated with user or group.
sd_element_flag	integer	R	Indicates whether to report the structured data element at runtime for this activity. The value at a particular index position applies to the structured data element identified in sd_element_name property at the corresponding index position. Valid values are: <ul style="list-style-type: none"> • 0: Do not report (default). • 1: Report.

Property	Datatype	Single or repeating	Description
sd_element_name	string(255)	R	<p>Names of the structured data elements that are visible to the activity.</p> <p>The names must be a subset of the structured data element names specified in dm_process.sd_element_name.</p>
sign_off_required	Boolean	S	<p>Specifies whether the activity requires a signoff. If this is set to T, the user or application must execute a Signoff method to complete the activity.</p> <p>The default is F (FALSE).</p>

Property	Datatype	Single or repeating	Description
task_subject	string(255)	S	<p>A user-defined string describing the task generated for this activity. The string may contain key phrases that are interpreted at runtime. The key phrases have the format:</p> <p><code>type.property</code></p> <p>Type can have the one of the following values:</p> <ul style="list-style-type: none"> • <code>dm_workflowdmi_workitem</code> • <code>dmi_queue_itemd mi_package</code> • <code>dmi_package[x]</code> <p>If type is <code>dm_workflow</code>, <code>dmi_workitem</code> or either package option, any property type can be specified.</p> <p>If type is <code>dmi_queue_item</code>, any queue item property except <code>task_subject</code> can be specified.</p>

Property	Datatype	Single or repeating	Description
transition_eval_cnt	integer	S	<p>Defines how many tasks must be completed to trigger activity completion.</p> <p>This property is not valid for activities with the following performer type:</p> <p>9: Some Users Sequentially.</p> <p>Valid values are 0 and any positive integer value. Setting this value to 0 requires that all tasks must be completed before the activity is completed. The default value is 0.</p>

Property	Datatype	Single or repeating	Description
transition_flag	integer	S	<p>Determines which output links are actually triggered when an activity completes. This value is used only if transition_type is set to 1.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • 0: Trigger all selected output links. • 1: Trigger only revert links if any are selected and ignore forward links. If there are no revert links selected, trigger selected forward links. • 2: Trigger only forward links if any are selected and ignore any selected revert links. If there are no forward links selected, trigger revert links. • 3: Trigger a revert link immediately if one is selected. • 4: Trigger a forward link immediately if one is selected.

Property	Datatype	Single or repeating	Description
transition_max_output_cnt	integer	S	<p>Specifies the maximum number of output ports that a performer can select.</p> <p>This property is valid only when transition_type is set to 1.</p> <p>A valid value is 0 or any positive integer. 0 means that there are no limits on the number of output ports the user can select. The default is 0.</p>
transition_type	integer	S	<p>Specifies the type of the transition condition. Valid values are:</p> <ul style="list-style-type: none"> • 0: Prescribed. Packages are forwarded to output ports when the post-condition is met. • 1: Manual. Activity performers specify the output ports with the Setoutput method. • 2: Automatic. Conditional routing logic computes the output ports. <p>The default value is 0.</p>
trigger_event	string(64)	S	Contains the expected event type that can trigger this activity instance.

Property	Datatype	Single or repeating	Description
trigger_threshold	integer	S	Specifies the threshold of the trigger condition; any number between 1 and the total number of input ports. The default value is 0. The default value is only allowed for begin activities, which have no input ports.

3.5 Activity group instance

Specifies a workflow activity group instance.

3.5.1 Description

- Supertype:
- Internal name: dmc_act_group_instance
- Object type tag: 00

3.5.2 Properties

The table describes the object properties.

Table 3-4: Activity group instance type properties

Property	Datatype	Single or repeating	Description
act_group_flag	integer	R	Contains the activity group flag of the activity. The flag at each index position corresponds to the activity identified at the corresponding index position in act_name.
act_group_id	ID	S	ID of the activity group (dm_activity object).
act_group_name	string(128)	S	Name of the activity group.

Property	Datatype	Single or repeating	Description
act_group_sequenceno	integer	S	The unique sequence number of the activity group instance.
act_group_state	integer	S	State of the activity group instance. Valid values are: <ul style="list-style-type: none"> • 0: Dormant • 1: Active • 2: Finished
act_name	string(128)	R	Contains the activity identifier as defined in the dm_process object.
act_sequenceno	integer	R	Contains the unique sequence number. The flag at each index position corresponds to the activity identified at the corresponding index position in act_name.
act_state	integer	R	Records the activity's current state. Valid values are: <ul style="list-style-type: none"> • 0: Dormant • 1: Active • 2: Finished
end_date	Time	S	Represents the end time of the activity group instance; set by the server when the workflow exits the activity group.
start_date	Time	S	Represents the start time of the activity group instance; set by the server when the workflow enters the activity group.
workflow_id	ID	S	ID of the workflow object (dm_workflow) to which the activity instance belongs.

3.6 Aggr domain

An aggr domain object describes the set of properties in an object type for which data dictionary information is defined.

3.6.1 Description

- Supertype: Domain
- Domain Subtypes: None
- None Internal name: dm_aggr_domain
- Object type tag: 51

Aggr domain objects are created and managed by the server. Users cannot create an aggr domain object.

3.6.2 Properties

The table describes the object properties.

Table 3-5: Aggr domain type properties

Property	Datatype	Single or repeating	Description
attr_domain_id	ID	R	Object IDs of the domains that describe the properties.
attr_domain_name	string(32)	R	Currently unused.
attr_name	string(32)	R	The names of properties that have data dictionary information in their definitions.
code_objects	ID	R	Object IDs of the expr code objects holding source code that implements the expressions associated with the type, if any. Currently, only one value is allowed in this property.
i_dd_flags	integer	R	Used internally.

Property	Datatype	Single or repeating	Description
type_name	string(32)	S	Name of the object type to which the properties belong. This value is NULL for lifecycle state or other non-type overrides.

3.7 Alias set

Records aliases and their values.

3.7.1 Description

- Supertype: Persistent
- Object Subtypes: None
- Internal name: dm_alias_set
- Object type tag: 66

An alias set object contains one or more alias names that can be used as placeholders in template ACLs, in certain SysObject properties, in workflow definitions, and in Link and Unlink methods. Any user can create alias sets.

3.7.2 Properties

The table describes the object properties.

Table 3-6: Alias set type properties

Property	Datatype	Single or repeating	Description
alias_category	integer	R	Specifies the category of the value defined at the corresponding index position in alias_value. Valid values and their associated categories are: <ul style="list-style-type: none"> • 0: Unknown • 1: User • 2: Group • 3: User or Group • 4: Cabinet path • 5: Folder path • 6: ACL name
alias_description	string(255)	R	User-defined description of the alias name at the corresponding index position in the alias_name property.
alias_name	string(32)	R	List of alias for the alias set. An alias_name value cannot contain a period (.)
alias_usr_category	integer	R	User-defined category values for the values in the alias_value property.
alias_value	string(255)	R	The actual user or group name associated with the alias that appears at the corresponding index level in the alias_name property.
object_description	string(128)	S	User-defined description of the alias set.

Property	Datatype	Single or repeating	Description
object_name	string(32)	S	Name of the alias set. The name must be unique among all alias set objects in the repository and cannot contain a period (.) or be composed entirely of numerical digits.
owner_name	string(255)	S	Name of the user who owns the alias set.

3.8 App ref

An app ref object is used internally to reference an object used in an application.

3.8.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_app_ref
- Object type tag: 07

Objects of this type reference objects used in a DAR file. Users cannot create app ref objects directly. They are created and managed by Documentum CM Server.

3.8.2 Properties

The table describes the object properties.

Table 3-7: App ref type properties

Property	Datatype	Single or repeating	Description
application_obj_id	ID	S	Object ID of the object referenced by the application.

3.9 Application

An application object represents a packaged application.

3.9.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_application
- Object type tag: 08

Application objects represent packaged DAR files. Users cannot create application objects directly. They are created when a user generates a DAR file.

3.9.2 Properties

The table describes the object properties.

Table 3-8: Application type properties

Property	Datatype	Single or repeating	Description
app_version	string(16)	S	Application version label.
application_object_id	ID	R	Object IDs of the objects in the application.

Property	Datatype	Single or repeating	Description
content_transfer_option	integer	R	<p>Defines how to handle cabinets or folders when copying them to a target repository. Valid values are:</p> <ul style="list-style-type: none"> • 0: Copy all contained objects. • 1: Copy only the hierarchy, the directly or indirectly contained folders, but not the documents in the folders. • 2: Copy just the cabinet or folder itself, but none of its contained objects. • 3: Copy only the cabinet or folder and its directly contained documents. <p>The value at an index position applies to the object at the corresponding index position in the application_object_id property.</p>
copyright_string	string(255)	S	User-defined.
def_alias_set_id	ID	S	The object ID of the default alias set for the application.
post_install_proc_id	ID	S	The chronicle ID of the post-installation procedure.
post_install_proc_label	string(32)	S	The version label of the post-installation procedure.
pre_install_proc_id	ID	S	The chronicle ID of the pre-installation procedure.

Property	Datatype	Single or repeating	Description
pre-install_proc_label	string(32)	S	The Version label of the pre-installation procedure.
target_loc_alias	string(64)	R	An alias that resolves to the location to which to copy the object in the application. The value at an index position applies to the object at the corresponding index position in application_object_id .
target_perm_alias	string(64)	R	An alias that resolves to the name of a template ACL. The value at an index position applies to the object at the corresponding index position in the application_object_id property.
target_owner_alias	string(64)	R	An alias that resolves to the name of the owner of the object. The value at an index position applies to the object at the corresponding index position in the application_object_id property.

Property	Datatype	Single or repeating	Description
upgrade_option	integer	S	<p>Specifies how to handle the object when the application is upgraded. The value at an index position applies to the object at the corresponding index position in the application_object_id property. Valid values are:</p> <ul style="list-style-type: none"> • 0: Overwrite the object. • 1: Ignore the object. • 2: Version the object.
user_runnable	Boolean	S	Reserved for future use.

3.10 Aspect relation

Describes the compatibility of two aspects.

3.10.1 Description

- Supertype: Relation
- Subtypes: None
- Internal name: dmc_aspect_relation
- Object type tag: 37



Caution

This object type is for internal use only. It is currently not supported for external use.

An aspect relation object type describes the compatibility of two aspects.

3.10.2 Properties

The table describes the object properties.

Table 3-9: Aspect relation type properties

Property	Datatype	Single or repeating	Description
child_aspect_name	string(64)	S	Reserved for internal use.
parent_aspect_name	string(64)	S	Reserved for internal use.
relation	integer	S	Reserved for internal use.

3.11 Aspect type

Stores the files that implement an aspect.

3.11.1 Description

- Supertype: Module
- Subtypes: None
- Internal name: dmc_aspect_type
- Object type tag: 0b

An aspect type object is a folder that contains the implementation classes and other information about one aspect. An aspect is a custom behavior implemented for a particular object type class. Aspect type folders are stored in the repository under System/Business Objects/Aspects.

3.11.2 Properties

The table describes the object properties.

Table 3-10: Aspect type object type properties

Property	Datatype	Single or repeating	Description
aspect_category	string(32)	R	Used internally.
attaching_aspect_policy	string(128)	R	Used internally.
copy_policy	integer	S	Used internally.
detaching_aspect_policy	string(128)	R	Used internally.

Property	Datatype	Single or repeating	Description
i_attr_def	string(27)	S	Used internally.
immutability_policy	integer	S	Used internally.
is_data_shared	Boolean	S	Used internally.
target_object_type	string(27)	R	Used internally.
version_policy	integer	S	Used internally.

3.12 Assembly

An assembly object describes a component of a virtual document.

3.12.1 Description

- Supertype: Persistent
- Object Subtypes: None
- Internal name: dm_assembly
- Object type tag: 0d

Assembly objects are created when a user creates an assembly, a snapshot of the virtual document at particular point in time. Users must have at least Version permission for the object identified in the book_id property to modify an assembly object.

3.12.2 Properties

The table describes the object properties.

Table 3-11: Assembly type properties

Property	Datatype	Single or repeating	Description
a_contain_desc	string(255)	S	Used by OpenText Documentum CM clients to manage XML documents in assemblies. <i>OpenText Documentum Content Management - Server Fundamentals Guide (EDCCS250400-GGD)</i> contains more information.

Property	Datatype	Single or repeating	Description
a_contain_type	string(32)	S	Used by OpenText Documentum CM clients to manage XML documents in assemblies. <i>OpenText Documentum Content Management - Server Fundamentals Guide (EDCCS250400-GGD)</i> contains more information.
book_id	ID	S	The object ID of the topmost document in the virtual document structure that contains the component described by this assembly object.
component_chron_id	ID	S	The chronicle ID of the component represented by the assembly object.
component_id	ID	S	The object ID of the component of the virtual document.
depth_no	integer	S	The component depth within the levels of the virtual document identified by the book_id.
i_partition	integer	S	Specifies the partition where the object resides.
order_no	integer	S	The ordering of the component within the virtual document identified by the book_id.
parent_id	ID	S	The object ID of the document that directly contains the virtual document described by this assembly object.

Property	Datatype	Single or repeating	Description
path_name	Oracle: string(740) SQLServer: string(765) PostgreSQL: string(200)	S	The path from the top-most document to the component. Path names are comprised of the node object names delimited by forward slashes. If the path is longer than the property, the path is truncated from the end of the path.

3.13 Atmos store

Represents an Atmos content storage area.

3.13.1 Description

- Supertype: Store
- Subtypes: none
- Internal name: dm_atmos_store
- Object type tag: inherited from Store

An Atmos store object represents an Atmos content storage area. The properties in Atmos store objects may be changed only by users with Sysadmin or Superuser user privileges.

3.13.2 Properties

Atmos store objects inherit all of their properties from their supertype, the Store type.

3.14 Attachments folder

Used to organize email message attachments.

3.14.1 Description

- Supertype: Folder
- Subtypes: None
- Internal name: dm_attachments_folder
- Object type tag: 0b

The attachments folder object organizes the email attachments stored in a repository.

3.14.2 Properties

Attachments folders inherit all of the properties from their Folder supertype.

3.15 Audit trail

An audit trail object stores information about one audited event.

3.15.1 Description

- Supertype: Persistent
- Subtypes: Audittrail Acl, Audittrail Group
- Internal name: dm_audittrail
- Object type tag: 5f

Audit trail objects are created automatically by the server after a user initiates auditing for a particular event. Automatic auditing is provided for system events, such as a checkin or checkout, workflow events, and lifecycle events.

Users and applications can also create audit trail objects to record the occurrence of user-defined events.

Audit trail objects have ten properties that dependent on the event. These properties are the `id_n` and `string_n` properties. For more information about auditable events, `id_n`, and `string_n` properties for server-generated audit trail objects, see *OpenText Documentum Content Management - Administrator User Guide (EDCAC250400-UGD)*.

3.15.2 Properties

The table describes the object properties.

Table 3-12: Audit trail type properties

Property	Datatype	Single or repeating	Description
acl_domain	string(255)	S	The owner of the ACL associated with the object being audited. This is null for objects that are not SysObjects or SysObject subtypes.
acl_name	string(32)	S	The name of the ACL associated with the object being audited. This is null for objects that are not SysObjects or SysObject subtypes.
application_code	string(64)	S	The application code or codes defined for the session in which the event occurred. Multiple codes are separated by commas. The list of codes is truncated if necessary.

Property	Datatype	Single or repeating	Description
attribute_list	string(2000)	S	<p>The comma-separated list of audited properties and their values at the time of the event. If the list is longer than the length of the property, the overflow is stored in the dmi_audittrailAttrs object identified in the attribute_list_id property.</p> <p>In case, event_name is dm_audit or dm_unaudit, then it stores the details of dmi_registry information provided during auditing.</p>
attribute_list_id	ID	S	<p>The object ID of the dmi_audittrailAttrs object that contains the overflow values from the attribute_list property.</p>
attribute_list_old	string(1000)	S	<p>The comma separated list of audited properties and their old values if the values changed.</p> <p>If the list is longer than the length of the property, the overflow is stored in the dmi_audittrailAttrs object identified in the attribute_list_id property.</p>
audit_signature	string(255)	S	The signed hash of the audit trail entry.

Property	Datatype	Single or repeating	Description
audit_version	integer	S	Specifies how the signature was created if the entry is signed. Entries created prior to the 5.2 release do not have this attribute set. Entries created from release 5.2 until release 6.5 set this value to 1. Entries created from release 6.5 until release 6.6 set this value to 2. Entries created from release version 6.6 and later, set the value to 4.
audited_obj_id	ID	S	The object ID of the audited object. For a checkin event, this is the object ID of the new version of the object. For workflow events, this is always the object ID of the process object. For dm_move_content events, this is always the object ID of the content object.
audited_obj_vstamp	integer	S	The i_vstamp value of the object identified in the audited_obj_id property.
chronicle_id	ID	S	The chronicle ID of the audited object if the object is a SysObject or SysObject subtype. Otherwise, this is the object ID of the audited object.

Property	Datatype	Single or repeating	Description
controlling_app	string(32)	S	The application controlling the object. This is the value in the a_controlling_app property of the audited object.
current_state	string(32)	S	The name of the current lifecycle state of the audited object.
event_description	string(64)	S	The name of the audited event in a user-friendly form.
event_name	string(64)	S	Actual name of the audited event.
event_source	string(64)	S	The source of the event. Some possible values are lifecycle, electronic signature, system, unspecific, workflow.
host_name	string(128)	S	The name of the Documentum CM Server host that generated the audit trail entry.
i_audited_obj_class	integer	S	<p>Specifies what type of object is identified in audited_obj_id. Values are:</p> <ul style="list-style-type: none"> • -1: The event has no audited object (for example, dm_audit or dm_unaudit events) • 0: dm_sysobject and subtypes • 1: dm_acl • 2: dm_user and subtypes • 3: dm_group • 4: dmr_content

Property	Datatype	Single or repeating	Description
i_is_archived	Boolean	S	<p>Specifies whether the audit trail entry is archived. Possible values are:</p> <ul style="list-style-type: none"> • <i>T</i>: The audit trail entry has been archived. • <i>F</i>: The entry has not been archived. <p>The audit trail entry is created and reset to <i>T</i> when the MARK_AS_ARCHIVED administration method is executed against the entry.</p>
id_1	ID	S	Event-dependent object ID.
id_2	ID	S	Event-dependent object ID.
id_3	ID	S	Event-dependent object ID.
id_4	ID	S	Event-dependent object ID.
id_5	ID	S	Event-dependent object ID.
object_name	string(255)	S	The name of the audited object. For SysObjects and ACLs, this is the object_name property value. For groups, this is the group_name value.
object_type	string(32)	S	The object type of the audited object.
owner_name	string(255)	S	The name of the owner of the audited object. The value can be an individual user or a group name.
policy_id	ID	S	The object ID of the lifecycle associated with the audited object.

Property	Datatype	Single or repeating	Description
r_gen_source	integer	S	The source of the audit trail. Valid values are: <ul style="list-style-type: none"> • 1: System-generated audit trail. • 0: User-generated audit trail.
r_object_sequence	integer	S	Indicates the creation sequence for dm_auditral objects. Reserved for future use. <p> Note: The usage of this attribute may cause integer overflow when sequence limit is exceeded. For instructions to reset the sequence counter for the supported databases, see the <i>Known issues</i> section in the product <i>Release Notes</i>.</p>
registry_id	ID	S	The object ID of the registry object that stores the registration for the event that generated this audit trail entry.
session_id	ID	S	The ID of the session in which the event occurred.
string_1	string(200)	S	Event-dependent string.
string_2	string(200)	S	Event-dependent string.
string_3	string(200)	S	Event-dependent string.

Property	Datatype	Single or repeating	Description
string_4	string(200)	S	Event-dependent string.
string_5	string(200)	S	Event-dependent string.
time_stamp	date	S	The local time at which the audit trail was generated.
time_stamp_utc	date	S	The UTC (or GMT) time at which the event occurred.
user_id	ID	S	The object ID of the dm_user object representing the user whose task caused the event to occur.
user_name	string(255)	S	The name of the user whose task caused the event to occur.
version_label	string(16)	S	The implicit (numeric) version label of the audited object. This is null for objects that are not SysObjects or SysObject subtypes.
workflow_id	ID	S	The object ID of the dm_workflow for workflow events.

Table 3-13: Additional audit trial type properties for external user transactions

Property	Datatype	Single or repeating	Description
audit_digest	string(255)	S	Encrypted checksum of all the stored data.
week_of_year	integer	S	Week of the year at the time of the transaction.
event_priority	integer	S	Reserved for future use.
transaction_id	string(255)	S	Reserved for future use.

3.16 Audit trail ACL

An audit trail acl object records information about one audited event for an ACL.

3.16.1 Description

- Supertype: Audit Trail
- Subtypes: None
- Internal name: dm_audittrail_acl
- Object type tag: 5f

Audit trail acl objects are created when a dm_save, dm_saveasnew, or dm_destroy registered event occurs on an ACL. The audited ACL is identified by the inherited audited_obj_id property.

3.16.2 Properties

The table describes the object properties.

Table 3-14: Audit trail acl type properties

Property	Datatype	Single or repeating	Description
accessor_name	string(255)	R	The ACL user or group entry on which the operation at the corresponding index position in accessor_operation was performed.
accessor_operation	string(1)	R	The operation performed on the user or group entry identified in the accessor_name property. Valid values are: <ul style="list-style-type: none">• <i>U</i>: Update• <i>I</i>: Insert• <i>D</i>: Delete

Property	Datatype	Single or repeating	Description
accessor_permit	integer	R	The r_accessor_permit value of the audited ACL. The audited ACL is specified in the dm_audittrail_acl.accessor_name property.
accessor_xpermit	integer	R	The r_accessor_xpermit value of the audited ACL. The audited ACL is specified in the dm_audittrail_acl.accessor_name property at the corresponding index position.
acl_class	integer	S	The acl_class value of the audited ACL.
application_permit	string(128)	R	The application_permit value of the audited ACL. The audited ACL is specified in the dm_audittrail_acl.accessor_name property at the corresponding index position.
description	string(128)	S	The description value of the audited ACL.
globally_managed	Boolean	S	The globally_managed property value of the audited ACL.
is_group	Boolean	R	The r_is_group value of the audited ACL. The audited ACL is specified in the dm_audittrail_acl.accessor_name property at the corresponding index position.
is_internal	Boolean	S	The r_is_internal value of the audited ACL.

Property	Datatype	Single or repeating	Description
permit_type	integer	R	The r_permit_type value of the audited ACL. The audited ACL is specified in the dm_audittrail_acl.accessor_name property at the corresponding index position.

3.17 Audit trail Attrs

An audit trail attrs object stores the overflow property name and value pairs from the attribute_list property of an audit trail object.

3.17.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_audittrail_attrs
- Object type tag: 00

Documentum CM Server creates audit trail attr objects automatically. They record overflow property names and their values if the list of recorded properties is too long to fit into the attribute_list property of the associated audit trail object.

3.17.2 Properties

The table describes the object properties.

Table 3-15: Audit trail attrs type properties

Property	Datatype	Single or repeating	Description
attribute_list	string(2000)	R	The comma-separated list of audited properties and their values at the time of the event. If the list is longer than the length of the property, the overflow is stored in the dmi_audittrail_attrs object identified in the attribute_list_id property. In case, event_name is dm_audit or dm_unaudit, then it stores the details of dmi_registry information provided during auditing.
attribute_list_old	string(1000)	R	A comma-separated list of audited properties and their old values if the values changed.
audit_obj_id	ID	S	The object ID of the associated audit trail object.

3.18 Audit trail group

An audit trail group object records information about an audited event for a group.

3.18.1 Description

- Supertype: Audit Trail
- Subtypes: None
- Internal name: dm_audittrail_group
- Object type tag: 5f

Audit trail group objects are created when a dm_save, saveasnew, or dm_destroy registered event occurs on a group. The audited group is identified in the inherited audited_obj_id property.

3.18.2 Properties

The table describes the object properties.

Table 3-16: Audit trail group type properties

Property	Datatype	Single or repeating	Description
alias_set_id	ID	S	The alias_set_id property value of the audited group.
description	string(128)	S	The description property value of the audited group.
globally_managed	Boolean	S	The globally_managed property value of the audited group.
group_admin	string(255)	S	The user or group allowed to modify the audited group.
group_address	string(80)	S	The group_address property value of the audited group.
group_class	string(32)	S	The group_class property value of the audited group.
group_source	string(16)	S	Specifies the source of the audited group. This property is deprecated.
groups_names	string(255)	R	The names of groups that are users in the audited group. The operation is recorded in the groups_names_operation property in the corresponding index position.

Property	Datatype	Single or repeating	Description
groups_names_operation	string(1)	R	The operation that occurred on the group identified in the groups_name property at the corresponding index position. Valid values are: <ul style="list-style-type: none">• <i>U</i>: Update• <i>I</i>: Insert• <i>D</i>: Delete
is_dynamic	Boolean	S	Specifies whether the audited group is dynamic. Possible values are: <ul style="list-style-type: none">• <i>T</i> (TRUE): The audited group is a dynamic group.• <i>F</i> (FALSE): The audited group is not a dynamic group.
is_dynamic_default	Boolean	S	Controls whether eligible users become members of the audited group by default when they connect to the repository. Possible values are: <ul style="list-style-type: none">• <i>T</i> (TRUE): Users are added to the group.• <i>F</i> (FALSE): Users are not added to the group when they connect.
is_private	Boolean	S	The is_private property value of the audited group.

Property	Datatype	Single or repeating	Description
users_names	string(255)	R	The names of individual users in the audited group on which an operation occurred. The operation is recorded in the groups_names_operation property in the corresponding index position.
users_names_operation	string(1)	R	The operation that occurred on the user identified in the users_name property at the corresponding index position. Valid values are: <ul style="list-style-type: none"> • <i>U</i>: Update • <i>I</i>: Insert • <i>D</i>: Delete

3.19 Auth config

An auth config object contains the names of the primary and backup domain controllers for a particular domain.

3.19.1 Description

- Supertype: Persistent object
- Subtypes: None
- Internal name: dm_auth_config
- Object type tag: 00

The auth config object values are used when Documentum CM Server authenticates repository users on a Linux platform against a Windows domain. A repository can have only one auth config object. The administrator creates the object and when defining the unix_domain_used value in the auth_protocol property of the docbase config object.

3.19.2 Properties

The table describes the object properties.

Table 3-17: Auth config type properties

Property	Datatype	Single or repeating	Description
domain_name	string (32)	R	The name of a Windows domain.
primary_controller	string(32)	R	The name of the machine that is the primary domain controller for the domain specified at the corresponding index position in the domain_name property.
backup_controller	string(32)	R	The name of the machine that is the backup domain controller for the domain named at the corresponding index position in the domain_name property.

3.20 BLOB store

A BLOB store object represents a BLOB storage area.

3.20.1 Description

- Supertype: Store
- Subtypes: None
- Internal name: dm_blobstore
- Object type tag: 40

BLOB store objects represent BLOB store storage areas. Content stored in BLOB stores is stored in tables in the repository. A BLOB store storage area can store a maximum of 64 Kbytes of data.

3.20.2 Properties

The table describes the object properties.

Table 3-18: BLOB store type properties

Property	Datatype	Single or repeating	Description
ascii	Boolean	S	Specifies whether the contents are ASCII strings or arbitrary sequences of 8-bit characters.
selectable	Boolean	S	Not currently used.

3.21 Branch Office Caching Services config

Specifies the configuration of a OpenText™ Documentum™ Content Management Branch Office Caching Services server.

3.21.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_bocs_config
- Object type tag: 08

A Branch Office Caching Services config object defines the configuration of a Branch Office Caching Services server. All Branch Office Caching Services config objects reside in the global registry, in the /System/NetworkLocation folder. The objects are created and managed using Documentum Administrator.

3.21.2 Properties

The table describes the object properties.

Table 3-19: Branch Office Caching Services config type properties

Property	Datatype	Single or repeating	Description
base_url	string(240)	R	The base URL for accessing the Branch Office Caching Services server using the protocol identified at the corresponding index position in the supported_protocol property.
delivery_type	string(4)	S	<p>Specifies whether the Messaging Service server can send messages to the Branch Office Caching Services server or must wait for the Branch Office Caching Services server to pull messages from the Messaging Service server. Valid values are:</p> <ul style="list-style-type: none"> • <i>push</i>: Messaging Service can send messages. • <i>pull</i>: Messaging Service must wait for Branch Office Caching Services to pull messages. <p>The default value is <i>push</i>.</p>
docbase_names	string(255)	R	<p>A list of repository names.</p> <p>The list is either an inclusion list or exclusion list depending on the value of <i>is_inclusion_list</i>.</p>
external_url	string(240)	R	The Branch Office Caching Services external URL.

Property	Datatype	Single or repeating	Description
external_url_type	string(32)	R	The Branch Office Caching Services external URL type. The only value currently supported is <i>ws_push</i> . This value indicates a proxy URL for the Branch Office Caching Services message URL. This URL used by Messaging Service to deliver messages to Branch Office Caching Services when Branch Office Caching Services is in push mode behind a proxy.
is_inclusion_list	Boolean	S	<p>Specifies whether the Branch Office Caching Services server can service the repositories listed in the docbase_names property. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i>(TRUE): The Branch Office Caching Services server can service the repositories. • <i>F</i> (FALSE): The Branch Office Caching Services server cannot service the repositories.

Property	Datatype	Single or repeating	Description
network_locations	string(80)	R	<p>The network identifiers of the network locations served by the Branch Office Caching Services server.</p> <p>The order of the identifiers in this property defines the relative closeness of the location to the Branch Office Caching Services server.</p>
rw_capability	integer	S	<p>Indicates whether the Branch Office Caching Services server is enabled or disabled and whether it is capable of asynchronous write operations. Valid values are:</p> <ul style="list-style-type: none">• 0: Disabled• 1: Enabled, but no asynchronous write operations.• 9: Enabled with asynchronous write capabilities operations.

Property	Datatype	Single or repeating	Description
server_major_version	string(16)	S	<p>Specifies the major version of the Branch Office Caching Services server. This attribute works in conjunction with the server_minor_version attribute. The “Attribute description” on page 96 and <i>OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS250400-AGD)</i> contain more information.</p>
server_minor_version	string(16)	S	<p>Specifies the minor version level of the Branch Office Caching Services server. This attribute works in conjunction with the server_major_version attribute. <i>OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS250400-AGD)</i> contains more information.</p> <p>For the valid values, see the server_major_version attribute.</p>
supported_protocol	string(6)	R	<p>The network protocols supported by the Branch Office Caching Services server.</p> <p>Values in this property must all be lowercase. For example, http or https.</p>

Table 3-20: Attribute description

server_major_version	server_minor_version	Description
1	NA	The Content Access options are limited to Read Only and None (disabled).
2	1	Compatible with Documentum CM Server version 6.5 SPx.
2	2	Compatible with Documentum CM Server version 6.6 to 6.6 (Patch 21).
2	3	Compatible with Documentum CM Server versions 6.6 (Patch 22) to 7.0. You must specify this value only when this Branch Office Caching Services server connects to an Accelerated Content Services server with a version of 2.3 (that is, version 6.6 Patch 22 and later).

3.22 Builtin expr

A builtin expr object stores information needed to execute built-in server functions.

3.22.1 Description

- Supertype: Expression
- Subtypes: None
- Internal name: dm_builtin_expr
- Object type tag: 54

Builtin expr objects store information needed to execute built-in server functions. Documentum CM Server creates and manages builtin expr objects.

3.22.2 Properties

The table describes the object properties.

Table 3-21: Builtin expr type properties

Property	Datatype	Single or repeating	Description
builtin_tag	integer	S	<p>Specifies which server function to invoke. Valid values are:</p> <ul style="list-style-type: none"> • 1: NULL • 2: NOW • 3: TODAY • 4: TOMORROW • 5: YESTERDAY • 6: USER <p>None of the functions require passed arguments.</p>

3.23 Business Workspace Type Definition

Stores the Business Workspace type name.

3.23.1 Description

- Supertype: dm_sysobject
- Subtypes: None
- Internal name: dm_bws_type_def
- Object type tag: 08

3.23.2 Properties

The table describes the object properties.

Table 3-22: Business Workspace Type Definition properties

Property	Datatype	Single or repeating	Description
icon_name	string(255)	S	Icon used for the Business Workspace type or Business Workspace template. This is a graphics object utilized by Admin Console and Smart View clients.
bws_folder_locations	string(455)	R	Folder location in the repository where Business Workspace instances can be created.
include_subfolders	boolean	R	Allows Business Workspaces to be created in subfolders of the specified folder locations. Used in conjunction with bws_folder_locations.
locales	string(10)	R	Lists the locale values available in the name_labels/ descriptions attributes in the corresponding index position. For example, en, ja, fr, and so on.
name_labels	string(255)	R	Stores the localized Business Workspace type name in the corresponding index position.
descriptions	string(400)	R	Stores the Business Workspace type description in the corresponding index position.

3.24 Business Workspace template

Stores the Business Workspace template name.

3.24.1 Description

- Supertype: dm_sysobject
- Subtypes: None
- Internal name: dm_bws_template
- Object type tag: 08

3.24.2 Properties

The table describes the object properties.

Table 3-23: Business Workspace template properties

Property	Datatype	Single or repeating	Description
template_state	integer	S	<p>Indicates the status of the template. Valid values are:</p> <ul style="list-style-type: none"> • 0 - draft • 1 - installed • 2 - noncurrent <p>Business Workspace instances can be created only from the installed version. At any given time, only one version is marked as installed. When the template is modified, the new version enters a draft state. Once published, the draft becomes the installed version, and the previous version is marked as noncurrent.</p> <p> Note: It is possible that the CURRENT version label can be the draft version from which Business Workspace cannot be created.</p>
bws_type_def_id	ID	S	Refers to the <code>r_object_id</code> of the corresponding <code>dm_bws_type_def</code> object for the template.

Property	Datatype	Single or repeating	Description
locales	string(10)	R	Stores the localized values as the repeating attributes. These values are available in the name_labels/descriptions attributes in the corresponding index position. For example, en, ja, fr, and so on.
name_labels	string(255)	R	Stores the Business Workspace template locale in the corresponding index position.
descriptions	string(400)	R	Stores the Business Workspace template description in the corresponding index position.

3.25 Business Workspace instance

Stores the Business Workspace instance name.

3.25.1 Description

- Supertype: dm_folder
- Subtypes: None
- Internal name: dm_bws
- Object type tag: 81

3.25.2 Properties

The table describes the object properties.

Table 3-24: Business Workspace Instance properties

Property	Datatype	Single or repeating	Description
template_id	ID	S	Indicates the r_object_id for dm_bws_template.

3.26 Business Workspace aspect

Stores the Business Workspace aspect.

3.26.1 Description

- Subtypes: None
- Internal name: dm_bws_aspect

3.26.2 Properties

The table describes the object properties.

Table 3-25: Business Workspace aspect properties

Property	Datatype	Single or repeating	Description
bws_id	ID	R	Indicates the r_object_id for dm_bws.
template_id	ID	R	Indicates the r_object_id for dm_bws_template.
node_id	string(100)	S	Unique ID associated to the folder during the design time. It is used as reference if the folder name or any other attributes is changed.

Property	Datatype	Single or repeating	Description
inherited_node_id	string(100)	S	When a folder is created manually inside a Business Workspace, the <code>inherited_node_id</code> attribute will be retained from the immediate parent. This ensures that rules configured for parent folder is applicable for this manual folder as well.

3.27 Ca store

A ca store object represents a retention-enabled storage system.

3.27.1 Description

- Supertype: Store
- Subtypes: None
- Internal name: dm_ca_store
- Object type tag: 6d

A retention-enabled storage system enforces retention policies at the device level, providing an additional level of content retention above that provided by Documentum CM Server. The object internal name, `dm_ca_store`, is retained for historical reasons, as the first supported retention-enabled storage system was a content-addressable storage system. A content addressable storage system uses a content address instead of a directory path to locate content stored in the system.

3.27.2 Properties

The table describes the object properties.

Table 3-26: Ca store type properties

Property	Datatype	Single or repeating	Description
a_content_attr_description	string(128)	R	The user-defined description of the <code>a_content_attr_name</code> property.

Property	Datatype	Single or repeating	Description
a_content_attr_name	string(64)	R	<p>The names of the content metadata fields. The metadata fields are specified using a Setcontentattrs method.</p> <p>The names may not contain spaces.</p> <p>This property can have a maximum of 62 values. Do not include the metadata field identified in a_retention_attr_name, if any.</p>
a_default_retention_date	Date	S	<p>The default retention value for content stored in this storage system.</p> <p>The value is ignored if</p> <ul style="list-style-type: none"> • a_retention_attr_required is set to T (TRUE) • a_retention_attr_name is set. • default_retention_days is set. <p>The default_retention_days is a property inherited from the dm_store object.</p>
a_plugin_id	ID	S	The object ID of the plugin object containing the plugin library that implements this storage system.

Property	Datatype	Single or repeating	Description
a_retention_attr_name	string(64)	S	<p>Specifies the content metadata field that stores the retention value for the content.</p> <p>The metadata field name cannot contain spaces. Do not use a metadata field defined in a_content_attr_name.</p> <p>The value of this property cannot be changed after the storage area is created.</p>
a_retention_attr_required	Boolean	S	<p>Specifies whether the content stored in the ca store requires a retention period. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i> (TRUE): The content stored in this system must have a retention period and the a_retention_attr_name property must be specified. • <i>F</i> (FALSE): The content stored in this system does not require a retention period. <p>The default is F (FALSE).</p>

Property	Datatype	Single or repeating	Description
a_storage_params	string(1024)	R	<p>The list of parameters specific to the retention-enabled storage system.</p> <p>For Centera systems, the first index position ([0]) is reserved for the IP address or addresses for the Centera host machine or machines. The value can include the path to a Centera profile. The second index position ([1]) is reserved for embedded BLOB configuration. If the index position is set, the value must have the the format:</p> <p><code>pool_option:embedded_blob:size_in_KB</code></p> <p>This format instructs the Centera system to store all content files less than or equal to the specified size as embedded BLOBs.</p> <p><i>OpenText Documentum Content Management - Administrator User Guide (EDCAC250400-UGD) contains more information.</i></p>

3.28 Cabinet

The cabinet object represents the highest level of organization visible to end users in a repository.

3.28.1 Description

- Supertype: Folder
- Subtypes: None
- Internal name: dm_cabinet
- Object type tag: 0c

The cabinet object represents the highest level of organization visible to end users in a repository. All folders, documents, and other objects except cabinets that are subtypes of the SysObject type are stored in cabinets. A cabinet is a folder that cannot be placed inside another folder or a cabinet. Users must have Superuser, Sysadmin, or Create Cabinet user privilege to create or destroy a cabinet. However, users can change cabinet properties if they have Write permission for the cabinet.

3.28.2 Properties

The table describes the object properties.

Table 3-27: Cabinet type properties

Property	Datatype	Single or repeating	Description
is_private	Boolean	S	<p>Indicates whether the cabinet is private or public. Valid values are:</p> <ul style="list-style-type: none"> • <i>TRUE</i>: The cabinet is private. • <i>FALSE</i>: The cabinet is public. <p>The default is <i>FALSE</i>. Client applications can use this property. Documentum CM Server does not use this property for security or any other purposes.</p>

3.29 Cache config

A cache config object identifies a group of queries, objects, or both to be cached on the client and refreshed at defined intervals.

3.29.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_cache_config
- Object type tag: 08

A cache config object identifies a group of queries, objects, or both to be cached on the client and refreshed at defined intervals.

3.29.2 Properties

The table describes the object properties.

Table 3-28: Cache config type properties

Property	Datatype	Single or repeating	Description
cache_element_query	string(1500)	R	The queries that return the query results, objects or both to be cached. The value can be any valid query. Each index position can contain one query.
cache_element_type	string(32)	R	Specifies whether the value at the corresponding position in cache_element_query identifies a query or object to be cached. Valid values are: query object

Property	Datatype	Single or repeating	Description
client_check_interval	integer	S	<p>Specifies in seconds how often the client calls the server to obtain updated information about the cache config object.</p> <p>The default value is 0, meaning that the client requests updated information every time an API call specifies the cache config object.</p>
i_query_result_hash	string(100)	S	The last computed hash of the query results. This value is used to compare executions of the cached queries.
r_last_changed_date	Date	S	The time at which the server last validated the queries and found a change.
r_last_checked_date	Date	S	The time at which the server last validated the queries.
server_check_interval	integer	S	<p>The interval, in seconds, at which the server revalidates the queries in the cache configuration.</p> <p>The default value is 0, meaning that the server reruns all the queries every time an API call specifies the cache config object and the client_check_interval has expired.</p>

3.30 Category

Represents a taxonomy in a repository.

3.30.1 Description

- Supertype: Folder
- Subtypes: Taxonomy
- Internal name: dm_category
- Object type tag: 0b

A category object is a folder in the folder structure that represents a taxonomy in a repository. Every category object must be linked to at least one taxonomy object or other category object.

For more information about categories and taxonomies, see the Content Intelligence Services documentation.

3.30.2 Properties

The table describes the object properties.

Table 3-29: Category type properties

Property	Datatype	Single or repeating	Description
a_status	string(16)	S	Describes the development and deployment status of the category. Valid values are offline and online.
allowed_operations	string(32)	R	List of allowed operations for the category. Valid values are user_browse, user_subscribe, and user_subscribe_mandatory.
candidate_threshold	integer	S	Relevance level in percentage, at which a document is automatically routed for approval to the category owner. Valid values are integers from 1 to 100.

Property	Datatype	Single or repeating	Description
category_evidence	string(255)	R	Specifies other categories that are considered as evidence for this category. Values for this property are set using the Content Intelligence Client.
category_owner	string(255)	R	The owner of the category. Only one owner is currently supported.
child_count	integer	S	The number of categories linked directly to this category.
class_id	ID	S	The object ID of the category class to which this category belongs. The default value is the class_id property in the taxonomy.
description	string(255)	S	User-defined description of the category.

Property	Datatype	Single or repeating	Description
definition_type	string(32)	S	<p>Specifies the type of the definition. The definition is comprised of the values in the category_evidence, keyword_evidence, candidate_threshold, and on_target_threshold properties. These values determine which documents can be automatically assigned to the category.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • <i>none</i> – The category has no definition or the definition is to be ignored. Documents cannot be automatically assigned to the category. • <i>simple</i> – The category has a simple definition.
keyword_evidence	string(255)	R	<p>The list of keywords in document text considered to be evidence for the category.</p> <p>The values in this property are set using Content Intelligence Client.</p>
object_name	string(255)	S	The name of the category. The name must be unique among all categories within the category class.

Property	Datatype	Single or repeating	Description
on_target_threshold	integer	S	The relevance level in percentage at which a document is automatically assigned to the category. The value can be an integer number 1 to 100.
qualifiers	string(255)	R	<p>The list of initial conditions that a document must meet to qualify for testing against the category evidence.</p> <p>The values in this property are set using Content Intelligence Client.</p>
supported_language	string(32)	R	<p>The list of language codes for all supported languages. The entry at each position must have a category name in the specified language at the matching index position in translated_name.</p>
translated_name	string(255)	R	<p>The list of the category name in all supported languages. The name at each index position must be in the language specified by the language code in the matching index position in supported_language.</p>

3.31 Category assign

Records the assignment of an object to a particular category.

3.31.1 Description

- Supertype: Relation
- Subtypes: None
- Internal name: dm_category_assign
- Object type tag: 37

A category assign object records the assignment of an object to a particular category. Category assign objects are created automatically when an object is assigned or proposed for assignment to a category folder.



Note: The data dictionary label for this type is Category Assignment.

3.31.2 Properties

The table describes the defined and inherited object properties.

Table 3-30: Category assign type properties

Property	Datatype	Single or repeating	Description
active_assignment	Boolean	S	<p>Specifies whether the object represented by this category assign object is visible to users. T (TRUE) means the object is available to end users. The value is T when assign_type value is active and the assign_status value begins with <i>assigned</i>.</p> <p>F (FALSE) means the object is not available.</p>

Property	Datatype	Single or repeating	Description
assign_status	string(32)	S	<p>Describes the assignment operation. Valid values are:</p> <ul style="list-style-type: none"> • assigned_auto • assigned_manual • assigned_approved • assigned_approved • assigned_pending_remove • assigned_final • pending_assign • removed_auto • removed_manual • removed_approved • removed_final
assign_type	string(32)	S	Specifies whether the assignment took place during a test or production run. Valid values is textactive.
child_id	ID	S	The i_chronicle_id value of the assigned object.
document_id	ID	S	Object ID of the assigned object.
modifier	string(255)	S	Name of the user or process that last modified the assignment. For all automatic assignment operations, this value is CIS.
parent_id	ID	S	Object ID of the category to which the object is assigned.

Property	Datatype	Single or repeating	Description
permanent_link	Boolean	S	Specifies whether to maintain the assignment across versions of the assigned object. The default is F (FALSE).
prev_assign_status	string(32)	S	Describes the assigned object's previous assignment, if any. Valid values are: <ul style="list-style-type: none">• assigned_auto• assigned_manual• assigned_approved• assigned_pending_remove• assigned_final• pending_assign• removed_auto• removed_manual• removed_approved• removed_final
prev_modifier	string(255)	S	Name of the user or process that performed the object's previous assignment. For all automatic assignments, this value is CIS.
relation_name	string(32)	S	Name of the relationship between the assigned object and the category. The only legal value is dm_category_assign.
relevance	integer	S	The relevance number of the assigned object. Values range from 1 to 100, with 100 meaning fully relevant.

Property	Datatype	Single or repeating	Description
run_id	ID	S	<p>The object ID of the dm_docset_run object that identifies the run that classified and assigned the object.</p> <p>This property can be a NULL value.</p>

3.32 Category class

Stores the definition of a class of categories and the default property values and behaviors for the included categories.

3.32.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_category_clas
- Object type tag: 08

A category class object stores the definition of a class of categories and the default property values and behaviors for the included categories. Category class objects are created using Content Intelligence Client.

3.32.2 Properties

The table describes the defined and inherited object properties.

Table 3-31: Category class type properties

Property	Datatype	Single or repeating	Description
cat_evidence_conf	string(32)	S	<p>Default confidence level for all explicit (not propagated) category evidence for categories in this class. Valid values are:</p> <ul style="list-style-type: none"> • certain • high • medium • low • auxiliary • negative • off • <i>numeric</i> • <i>numeric</i> is any value from 0 to 100.
description	string(255)	S	User-defined description of the category class.
evidence_prop_conf	string(32)	S	<p>Default confidence level for all propagated category evidence for categories in this class. Valid values are:</p> <ul style="list-style-type: none"> • certain • high • medium • low • auxiliary • negative <p><i>numeric</i>, where <i>numeric</i> is any value from 0 to 100.</p>

Property	Datatype	Single or repeating	Description
evidence_prop_type	string(32)	S	<p>Default value for all automatic (propagated) category evidence generation for categories in the class. Valid values are:</p> <ul style="list-style-type: none"> • <i>off</i>, which turns off evidence propagation@<i>child</i>, which propagates evidence from any child to the parent category@<i>parent</i>, which propagates any evidence for the parent category to immediate children
implied_keyword_conf	string(32)	S	<p>Default keyword confidence level for all categories in this class. Valid values are:</p> <ul style="list-style-type: none"> • certain • high • medium • low • auxiliary • negative • off <p><i>numeric</i>: any number from 0 to 100.</p>
implied_keyword_style	string(64)	S	<p>Default implied keyword style for categories in this class. Valid values are combinations of the following:</p> <ul style="list-style-type: none"> • random • nostem

Property	Datatype	Single or repeating	Description
keyword_conf	string(32)	S	<p>Default keyword confidence level for all categories in the class. Valid values are:</p> <ul style="list-style-type: none"> • certain • high • medium • low • auxiliary • negative • off <p><i>numeric:</i> an integer from 0 to 100.</p>
keyword_style	string(64)	S	<p>Default keyword style for categories in this class. Valid values are combinations of:</p> <ul style="list-style-type: none"> • active • stem • phrase_order_exact
object_name	string(255)	S	<p>Category class name. The name must be unique among the categories that belong to the class.</p>
source	string(32)	S	<p>User-defined text identifying the source of the taxonomy and category objects in this class. For example, DCTM or Semio.</p>
target_attribute	string(32)	S	<p>Name of the property in a repository object type to be used for storing all document category assignments. The property must be a repeating property.</p>

3.33 Change record

Stores information used internally to ensure consistency of cached information.

3.33.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_change_record
- Object type tag: 33

A change record object stores information used internally to ensure consistency of cached information. There is one change record object for each repository.

3.33.2 Properties

The table describes the defined and inherited object properties.

Table 3-32: Change record type properties

Property	Datatype	Single or repeating	Description
cache_change_count	integer	S	Number of changes affecting the global cache of the server.
dd_change_count	integer	S	Number of times the data dictionary has been changed.
dormancy_change_count	integer	S	Number of times Documentum CM Server has been toggled between the dormant state and the active state.
group_change_count	integer	S	Number of times any group object has been changed.
reinit_change_count	integer	S	Number of times the main server thread (parent server) has been reinitialized.
sys_change_count	integer	S	Number of changes made to object types and format objects.

Property	Datatype	Single or repeating	Description
storage_change_count	integer	S	Number of times that a storage area has changed state.
type_change_count	integer	S	Number of times that a type in the global cache has been changed.
user_change_count	integer	S	Number of times any user object has been changed.

3.34 CI config

Stores Content Intelligence Services configuration options.

3.34.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_ci_config
- Object type tag: 08

A CI config object stores Content Intelligence Services configuration options. CI config objects are created when a repository is enabled for Content Intelligence Services. Enabling occurs when Content Intelligence Client is installed or the DAR file for Web Publisher is installed.

3.34.2 Properties

The table describes the defined and inherited object properties.

Table 3-33: CI config type properties

Property	Datatype	Single or repeating	Description
assign_as_attribute	Boolean	S	Specifies whether reflecting the category assignments as document properties is enabled in the repository.
assign_as_link	Boolean	S	Specifies whether assigning objects to folders is enabled in the repository.

Property	Datatype	Single or repeating	Description
auto_proc_enabled	Boolean	S	Specifies whether automatic assignment is enabled in the repository.
auto_user	string(255)	S	User account that is used to generate and process the automatic document processing queue.
auto_workflow	Boolean	S	Specifies whether automated object assignment initiates a confirmation workflow.
cat_owner_suggest	Boolean	S	Not currently used.
category_object_type	string(32)	S	Name of the object type used to manage categories. This is set to dm_category by default.
ci_client_enabled	Boolean	S	Specifies whether the Content Intelligence Client is enabled on the repository.
ci_enabled	Boolean	S	Specifies whether Content Intelligence functionality is enabled on the repository.
ci_server_host_dev	string(80)	S	Host on which the Content Intelligence Services development server resides.
ci_server_host_prod	string(80)	S	Host on which the Content Intelligence Services production server resides.

Property	Datatype	Single or repeating	Description
impersonate_assign	Boolean	S	T (TRUE) allows both automatic and manual document assignments to be performed using the designated superuser account. The superuser account is identified using Content Intelligence Client.
manual_proc_enabled	Boolean	S	Specifies whether manual processing is enabled on the repository.
manual_user	string(255)	S	User account that is used to generate and process the manual document processing queue.
manual_workflow	Boolean	S	Specifies whether manual object assignment requires a confirmation workflow.
root_admin_path	string(255)	S	Folder path of the root folder from which all administration information managed.
root_category_path	string(255)	S	Folder path of the root folder from which all categories and taxonomies are managed.
sync_mode	string(32)	S	Specifies how data synchronization with the Content Intelligence server occurs. The only valid value is auto.

Property	Datatype	Single or repeating	Description
taxonomy_object_type	string(32)	S	Name of the object type used to manage taxonomies in the repository. This is set to dm_taxonomy by default.

3.35 Class

Represents a model for a Smart Container.

3.35.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_class
- Object type tag: 08

A class object represents a model for a Smart Container. The content of the object is the XML representation of the Smart Container model. Every object of this type has an associated dm_logical_version aspect. This object type is installed with the Smart Container DAR file during repository configuration.

3.35.2 Properties

The table describes the defined and inherited object properties.

Table 3-34: Class type properties

Property	Datatype	Single or repeating	Description
class_category	string(255)	S	Category of the model
class_description	string(1024)	S	Description of the model
metamodel_id	ID	S	Object ID of the dmc_metamodel object from which the model represented by this class object was derived.

The table describes the default aspect properties for the class type.

Table 3-35: Class type default aspect properties

Property	Datatype	Single or repeating	Description
logical_version	string(32)	S	Supports logical versioning of persistent objects From the aspect named dm_logical_aspect.

3.36 Client registration

Records registration information about client applications.

3.36.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_client_registration
- Object type tag: 08

A client registration object records the identity of a particular Privileged DFC instance. Client registration objects are stored in the global registry, in the /System/Client Registration folder. They are protected by the dm_acl_registration ACL. This ACL grants Browse permission to dm_world and Delete permission to dm_owner and the admingroup group.

3.36.2 Properties

The table describes the defined object properties.

Table 3-36: Client registration type properties

Property	Datatype	Single or repeating	Description
client_id	string(40)	S	Unique identifier for the client instance.
host_name	string(128)	S	Name of host on which the client is installed.

Property	Datatype	Single or repeating	Description
is_validated	Boolean	S	Specifies whether the entry has been checked for authenticity. T means that the entry has been checked. F means that the entry has not been checked.
public_key_identifier	string(40)	S	Points to the public key certificate object that holds the public key for this client.

3.37 Client rights

Records the roles that a particular instance of a privileged client is allowed to use.

3.37.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_client_rights
- Object type tag: 08

A client rights object records the roles that a particular instance of a client application is allowed to use within a repository. There must be a client rights object in each repository in which a Privileged DFC instance will invoke the roles. Client rights objects are created using Documentum Administrator. You must have Superuser privileges to create a client rights object. The objects are stored in the / System/Client Rights folder, and are assigned the dm_acl_superusers ACL.

3.37.2 Properties

The table describes the object properties.

Table 3-37: Client rights type properties

Property	Datatype	Single or repeating	Description
allow_all_roles	Boolean	S	T means the client may use any role. Setting this to T is the equivalent of listing all roles in allowed_roles.

Property	Datatype	Single or repeating	Description
allow_all_priv_modules	Boolean	S	T means the client may use any role invoked by an installed module.
allowed_roles	string(255)	R	List of the roles that the client is allowed to use.
app_name	string(128)	S	Name of client application on host. Helps to identify the application if there are multiple ones on the host.
client_id	string(40)	S	Unique identifier for the client instance.
host_name	string(128)	S	Name of host on which the client is installed.
is_globally_managed	Boolean	S	T means that this client rights object is globally managed. A globally managed client rights object is shared by all members of a client rights domain. Changes to a globally managed client rights object should be made in the govern repository of the domain. Changes are allowed to globally managed client rights objects in a member repository, but those changes may be overwritten by the govern repository.

Property	Datatype	Single or repeating	Description
principal_auth_priv	Boolean	S	T means that the client is allowed to create sessions for users without user credentials. This enables Principal Mode support for a Foundation Java API instance.
public_key_identifier	string(40)	S	Points to the public key certificate object that holds the public key for this client.
server_trust_priv	Boolean	S	When set to true, this client is taken as part of a trusted Documentum CM Server domain, this gives the client direct access to content storage.

3.38 Client rights domain

Describes the properties of a client rights domain object.

3.38.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dm_client_rights_domain
- Object type tag: 70

A client rights domain object indicates a set of repositories that share the same set of dm_client_rights information. When the domain is active, the client rights objects among the repositories is synchronized to the values in the master repository for the domain, known as the govern repository.



Note: If you remove a member repository from the client rights domain object, the removal will not occur until the dm_PropagateClientRights job completes successfully.

3.38.2 Properties

The table lists the properties defined for the type.

Table 3-38: Client rights domain type properties

Property	Datatype	Single or repeating	Description
is_active	Boolean	S	T means that the domain will synchronize the client rights objects for the repositories in the domain. F means that the domain is inactive and the client rights objects will not be synchronized among the repositories.
last_refresh	Time	R	The last successful propagation of client rights objects to the domain member repositories.
login_name	string(338)	R	User name to use to login to a member repository. If the user is a domain user, login_name will include the domain name in the format, "domain\user".
member_docbase	string(255)	R	Name of a member repository in the domain.
object_name	string(255)	S	Name of the client rights domain object.
r_govern_docbase	string(255)	S	Specifies the master repository.

Property	Datatype	Single or repeating	Description
r_modify_date	Time	S	<p>The time of the last change that triggers propagation.</p> <p>Changes that trigger propagation are:</p> <ul style="list-style-type: none"> • add or remove member repository • change to a globally managed client rights object <p>Changes that do not trigger propagation are:</p> <ul style="list-style-type: none"> • change login name of existing repository

3.39 Completed workflow

Records information about completed workflows.

3.39.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmc_completed_workflow
- Object type tag: 00

A completed workflow object stores information from audit trail records about a completed workflow. The objects are created by the dm_WfReporting job and used by Webtop's aggregate workflow reporting tool. To fully populate the properties in these objects, you must be auditing all workflow events.

The object type is created by a script when Documentum CM Server is installed.

3.39.2 Properties

The table describes the object properties.

Table 3-39: Completed workflow type properties

Property	Datatype	Single or repeating	Description
active_duration	double	S	<p>Length of time the workflow was in the active state.</p> <p>This value is computed from the audit trail entries for the dm_startworkflow, dm_changestateworkflow, dm_finishworkflow, and dm_abortworkflow events.</p>
complete_date_utc	Date	S	<p>The date and time on which the workflow is finished or aborted.</p> <p>This value is taken from the audit trail entries for the dm_finishworkflow or dm_abortworkflow events.</p>
complete_type	integer	S	<p>Specifies whether the workflow was finished or aborted. Valid values are:</p> <ul style="list-style-type: none">• 0: The workflow was finished• 1: The workflow was aborted <p>The value is taken from the audit trail entries for the dm_finishworkflow or dm_abortworkflow events.</p>

Property	Datatype	Single or repeating	Description
component_id	ID	S	Object ID of the first component in the first package. The value is taken from audit trail entries for the dm_addpackage event.
component_name	string(80)	S	Name of the first component in the first package. The value is taken from audit trail entries for the dm_addpackage event.
creator_name	string(255)	S	Name of the creator of the workflow. The value is taken from the audit trail entries for the dm_createworkflow event.
paused_duration	double	S	Length of time the workflow was in the paused state. The value is computed by subtracting the value of active_duration from total_duration.
process_id	ID	S	Object ID of the template used to create the workflow. The value is taken from the audit trail entries for the dm_createworkflow event.
process_name	string(255)	S	Name of the workflow template used to generate the workflow instance.

Property	Datatype	Single or repeating	Description
start_date_utc	Date	S	Date and time of the start of the workflow. The value is taken from the audit trail entries for the dm_startworkflow event.
supervisor_name	string(255)	S	Name of the supervisor of the workflow. The value is taken from the audit trail entries for the dm_startworkflow and dm_changeworkflow supervisor events.
total_duration	double	S	Total length of the time the workflow existed. The value is computed by subtracting start_date_utc from complete_date_utc.
total_user_cost	double	S	Sum of the user_cost values for the completed work items in the workflow. The value is computed from the dmc_completed_workitem objects.
total_user_time	integer	S	Sum of the user_time values for the completed work items in the workflow. The value is computed from the dmc_completed_workitem objects.

Property	Datatype	Single or repeating	Description
workflow_id	ID	S	Object ID of the completed workflow. The value is taken from dm_createworkflow events.
workflow_name	string(128)	S	Name of the workflow. The value is taken from the audit trail entries for the dm_createworkflow event.

3.40 Completed workitem

Records information about completed work items in workflows.

3.40.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmc_completed_workitem
- Object type tag: 00

A completed workitem object stores information from audit trail records about a completed work item. The objects are created by the dm_WfReporting job and used by Webtop's aggregate workflow reporting tool. To fully populate the properties in these objects, you must be auditing all workflow events.

The object type is created by a script when Documentum CM Server is installed.

3.40.2 Properties

The table describes the object properties.

Table 3-40: Completed workitem type properties

Property	Datatype	Single or repeating	Description
act_name	string(128)	S	Name of the activity that generated the work item. The value is taken from the audit trail entries for dmd_startedworkitem events.
act_seqno	integer	S	Sequence number of the activity that generated the work item. The value is taken from the audit trail entries for dmd_startedworkitem events.
active_duration	double	S	Length of time that the work item was in the active state. The value is computed from the audit trail entries for the dm_selectedworkitem, dm_completedworkitem, and dm_changestateactivity events.
complete_action	integer	S	Specifies the action chosen by the user when completing the work item. Valid values are: <ul style="list-style-type: none">• -1, meaning unknown• 0, meaning forward• 1, meaning reject The value is derived from the audit trail entries for the dm_setoutput event.

Property	Datatype	Single or repeating	Description
complete_date_utc	Date	S	<p>Date and time at which the work item was completed.</p> <p>The value is taken from the audit trail entries for dm_completedworkitem events.</p>
complete_state	integer	S	<p>State of the work item when the workflow was finished or aborted. Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning dormant • 1, meaning acquired • 2, meaning finished
creation_date_utc	Date	S	<p>Date and time when the work item was created.</p> <p>The value is taken from the audit trail entries for dm_startedworkitem events.</p>
dormant_duration	double	S	<p>Length of time, in seconds, that the work item was in the dormant state.</p> <p>The value is derived from the audit trail entries for dm_startedworkitem and dm_selectedworkitem events.</p>

Property	Datatype	Single or repeating	Description
paused_duration	integer	S	<p>Length of time, in seconds, that the work item was in the paused state.</p> <p>The value is derived from the values for total_duration, dormant_duration, and active_duration.</p>
performer_name	string(255)	S	<p>Name of the task performer.</p> <p>The value is taken from the audit trail entries for the dm_startedworkitem event.</p>
process_id	ID	S	<p>Object ID of the workflow template used to create the workflow that generated this work item.</p> <p>The value is taken from audit trail entries for the dm_startedworkitem event.</p>
process_name	string(255)	S	<p>Name of the workflow template used to generate the workflow instance that contained this work item.</p> <p>The value is taken from audit trail entries for the dm_startedworkitem event.</p>

Property	Datatype	Single or repeating	Description
task_priority	integer	S	The final priority of the work item. The value is taken from the audit trail entries for the dm_startedworkitem and dm_changepriorityworkitem events.
total_duration	integer	S	Total length of time, in seconds, that the work item existed. This is computed using start_date_utc and complete_date_utc values.
user_cost	double	S	The user cost argument value specified in the Complete method. The value is taken from audit trail entries for the dm_completedworkitem event.
user_time	integer	S	Amount of time the user spent on the work item (specified as an argument to the Complete method). The value is taken from audit trail entries for the dm_completedworkitem event.
workflow_id	ID	S	Object ID of the workflow that generated the work item. The value is taken from audit trail entries for the dm_startedworkitem events.

Property	Datatype	Single or repeating	Description
workitem_id	ID	S	Object ID of the work item. The value is taken from audit trail entries for the dm_startedworkitem events.
wq_doc_profile	string(64)	S	Value of the a_wq_doc_profile property of the work item. The value is taken from audit trail entries for the dm_completedworkitem events.
wq_flag	integer	S	Value of the a_wq_flag property of the work item. The value is taken from audit trail entries for the dm_completedworkitem events.
wq_name	string(32)	S	Value of the a_wq_name property of the work item. The value is taken from audit trail entries for the dm_completedworkitem events.

3.41 Component

Represents a component, a set of related functionality that is used by client applications.

3.41.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_component
- Object type tag: 08

A component object represents a component, a set of related functionality that is used by client applications.

3.41.2 Properties

The table describes the object properties.

Table 3-41: Component type properties

Property	Datatype	Single or repeating	Description
build_technology	integer	S	The build technology used to create the component. Valid values are: 1, for ACX 16, for WIN 32 Exes 32, for Data module
com_class_id	string(38)	S	The component's COM class ID. This is required for desktop applications. It is not required for Web-based applications.
component_version	string(16)	S	The version of the component.
uniq_cont_ticket	string(128)	S	Used internally.

3.42 Composite predicate

Represents one route case condition for a workflow activity.

3.42.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmc_composite_predicate
- Object type tag: 00

A composite predicate object records a route case condition for an activity that has a transition type of automatic. Composite predicate objects are created internally when the addConditionRouteCase method is used to add route case definitions to an activity. These objects cannot be created directly.

3.42.2 Properties

The table describes the object properties.

Table 3-42: Composite predicate type properties

Property	Datatype	Single or repeating	Description
predicate_id	ID	R	Object IDs of transition condition objects representing the individual comparison expressions that make up the route case.
r_aspect_name	string(64)	R	Names of the TBO associated with the predicate.

3.43 Cond expr

Stores a list of one or more Boolean expressions that are called by IF conditions in a routine or containing expression.

3.43.1 Description

- Supertype: Func Expr
- Subtypes: Cond ID Expr
- Internal name: dm_cond_expr
- Object type tag: 56

A cond expr object stores a list of one or more Boolean expressions that are called by IF conditions in a routine or containing expression. Documentum CM Server creates and manages cond expr objects.

3.43.2 Properties

The table describes the object properties.

Table 3-43: Cond expr type properties

Property	Datatype	Single or repeating	Description
expression_list	ID	R	A list of object IDs representing dm_func_expr objects for expressions that return a Boolean result.

3.44 Cond ID expr

Stores a list of one or more Boolean expressions and a list of corresponding object IDs.

3.44.1 Description

- Supertype: Cond Expr
- Subtypes: None
- Internal name: dm_cond_id_expr
- Object type tag: 57

A cond id expr object stores a list of one or more Boolean expressions and a list of corresponding object IDs. The list of expressions is stored in the inherited expression_list property. The information stored in a cond ID expr object is part of the data dictionary. A cond id expr object is created when a routine or larger expression includes one or more if conditions that return an object ID. Documentum CM Server creates and manages cond id expr objects. Users cannot create them.

3.44.2 Properties

The table lists the properties defined for the type.

Table 3-44: Cond ID expr type properties

Property	Datatype	Single or repeating	Description
default_id	ID	S	Object ID to return if none of the expressions evaluate to TRUE.
id_list	ID	R	<p>Object IDs to be returned when the corresponding expression evaluates to TRUE.</p> <p>For example, if the expression specified in expression_list[3] evaluates to TRUE, the object ID specified in id_list[3] is returned.</p>

3.45 Config scope relation

Describes a configuration scope and relates a preset package object to its scope.

3.45.1 Description

- Supertype: Relation
- Subtypes: None
- Internal name: dmc_config_scope_relation
- Object type tag: 37

A config scope relation object is used to relate a preset package to the preset scope of the package. The properties of the config scope relation object record information about the scope. The name of the relationship represented by objects of type config scope relation is dmc_config_scope_rel_type. This object type is installed with the Preset DAR file during repository configuration.

3.45.2 Properties

The table describes the object properties.

Table 3-45: Config scope relation type properties

Property	Datatype	Single or repeating	Description
scope_name	string(32)	S	Name of the scope. This value is defined by the application.
scope_values	string(64)	R	Specifies the values to which the related preset configuration applies.

3.46 Constraint set

Records a collection of constraints applied to relationships.

3.46.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_constraint_set
- Object type tag: 08

A constraint set object records one or more predicates that define constraints applied to the parent or child in a relationship defined by a dmc_relationship_def object. Constraint set objects are created and managed internally. Users should not directly manipulate these objects. The object type is installed with the Smart Container DAR file during repository configuration.

3.46.2 Properties

The table describes the object properties.

Table 3-46: Constraint set object type properties

Property	Datatype	Single or repeating	Description
constraint_name	string(255)	R	Name of the constraint.

Property	Datatype	Single or repeating	Description
constraint_metadata	string(255)	R	Metadata with which to configure the module identified in module_name prior to invoking the module.
module_name	string(255)	R	Name of the BOF module that implements the constraint.

3.47 Cont transfer config

Controls the capabilities of the Accelerated Content Services servers that service the repository and whether content in the repository may be pre-cached on a Branch Office Caching Services server.

3.47.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_cont_transfer_config
- Object type tag: 08

A cont transfer config object defines a top-level of control for the read and write capabilities of the Accelerated Content Services servers associated with the repository. In addition, it controls whether content in the repository can be pre-cached on a Branch Office Caching Services server.

A cont transfer config object is created during repository installation. Although a Superuser is allowed to create an object of this type, there should be only one cont transfer config object in a repository.

3.47.2 Properties

The table describes the object properties.

Table 3-47: Cont transfer config type properties

Property	Datatype	Single or repeating	Description
acs_read_mode	integer	S	<p>Specifies whether Accelerated Content Services servers associated with the repository can transfer content to users for reading. Valid values are:</p> <ul style="list-style-type: none"> • 0: Disabled • 1: Enabled <p>The default is 1.</p> <p>The value in this property overrides any capability defined for individual Accelerated Content Services servers.</p>
acs_write_mode	integer	S	<p>Specifies whether Accelerated Content Services servers associated with the repository can transfer content from users to the repository. Valid values are:</p> <ul style="list-style-type: none"> • 0: Disabled • 1: Enabled • 3: Enabled for transfers and asynchronous write operations. • 7: Enabled for asynchronous write operations and retention. <p>The default is 0.</p> <p>The value in this property is disregarded for individual Accelerated Content Services servers that are disabled.</p>
bocs_precaching_mode	integer	S	<p>Indicates whether content in the repository may be pre-cached on a Branch Office Caching Services server or not. Valid values are:</p> <ul style="list-style-type: none"> • 0: Pre-caching content is not allowed • 1: Pre-caching content is allowed <p>The default is 1.</p>

Property	Datatype	Single or repeating	Description
bocs_encryption_mode	string(32)	S	<p>This attribute is the default value for the encryption_mode parameter of the Branch Office Caching Services content URL.</p> <p>If you use a default type-based object (TBO) and the content is in an encrypted store, <i>require</i> is used regardless of the value of this attribute. If you write your own TBO, the content will be encrypted on Branch Office Caching Services based on what the <code>doGetContentEncryptionMode</code> method returns, regardless of the value of this attribute.</p> <p>Valid values are:</p> <ul style="list-style-type: none">• <i>allow</i> - encrypt content on Branch Office Caching Services only if the <code>encryption.mode</code> parameter of the <code>acs.properties</code> file on Branch Office Caching Services is set to <i>require</i>.• <i>require</i> - encrypt content on Branch Office Caching Services. Content will not be stored on Branch Office Caching Services if the Branch Office Caching Services version does not support encryption or if the <code>encryption.mode</code> parameter of the <code>acs.properties</code> file on Branch Office Caching Services is set to <i>disable</i>.• <i>disable</i> - do not encrypt content on Branch Office Caching Services. <p>The default value is <i>allow</i>.</p>

3.48 Containment

Stores information about a component of a virtual document.

3.48.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmr_containment
- Object type tag: 05

A containment object stores information about a component of a virtual document. Each time a user adds a component to a virtual document, the server creates a containment object for that component. The properties for this type are set by the Appendpart, Insertpart, and Updatepart methods. Users can query these properties using DQL.

3.48.2 Properties

The table describes the object properties.

Table 3-48: Containment type properties

Property	Datatype	Single or repeating	Description
a_contain_desc	string(255)	S	User-defined. Used by OpenText Documentum CM clients to manage XML documents. <i>OpenText Documentum Content Management - Server Fundamentals Guide (EDCCS250400-GGD)</i> contains more information.
a_contain_type	string(255)	S	User-defined. Used by OpenText Documentum CM clients to manage XML documents. <i>OpenText Documentum Content Management - Server Fundamentals Guide (EDCCS250400-GGD)</i> contains more information.

Property	Datatype	Single or repeating	Description
component_id	ID	S	Chronicle ID of the component. The chronicle ID is the object ID of the original version of an object. If the object has no versions, then its object ID and chronicle ID are the same.
copy_child	integer	S	<p>Defines client behavior when the document containing the component is copied. Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning the decision whether to copy or reference the component is left to the user when the document is copied. • 1, meaning when the document is copied, the component is referenced in the new copy rather than actually copied. • 2, meaning when the document is copied, the component is also copied.
follow_assembly	Boolean	S	If set to TRUE, directs the system to resolve a component using the component's assembly (if the component has an assembly).
i_partition	integer	S	Specifies the partition where the object resides.

Property	Datatype	Single or repeating	Description
order_no	double	S	Number representing the component's position within the components of the virtual document identified by parent_id.
parent_id	ID	S	Object ID of the object that directly contains the component.
use_node_ver_label	Boolean	S	If set to TRUE for early-bound components, the server uses the early-bound symbolic label to resolve late-bound descendants of the component during assembly.
version_label	string(32)	S	Version label for the component.

3.49 Content

Stores information about a content file.

3.49.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmr_content
- Object type tag: 06

A content object stores the information about the format and location of a content file. It also contains the information that links the content to an object. A content object also has five properties used by Content Transformation Services to store metadata values generated by a Content Transformation Services server. If the object is stored in a content-addressed storage area, these properties are used to record metadata values to be stored in the storage system with the content.

3.49.2 Properties

The table describes the object properties.

Table 3-49: Content type properties

Property	Datatype	Single or repeating	Description
content_attr_data_type	integer	R	Data type of the media property in the corresponding index position in content_attr_name. Valid values are: <ul style="list-style-type: none">• 2, for a string datatype• 4, for a date/time datatype• 5, for a double datatype
content_attr_date_value	date	R	The value of the metadata field identified in the corresponding index position in content_attr_name if the property is a date data type. For all other data types, the value in the corresponding index position in content_attr_date_value is NULLDATE.
content_attr_name	string(64)	R	Names of the metadata fields generated by a Content Transformation Services server or specified in a Setcontentattrs or SET_CONTENT_AT TRS method for the content.

Property	Datatype	Single or repeating	Description
content_attr_num_value	integer	R	The value of the metadata field identified in the corresponding index position in content_attr_name if the property is an integer data type. For all other data types, the value in the corresponding index position in content_attr_num_value is NULLINT.
content_attr value	string(255)	R	The value of the metadata field identified in the corresponding index position in content_attr_name if the property is an string data type. For all other data types, the value in the corresponding index position in content_attr_value is NULLSTRING.
content_size	integer	S	<p>Size, in bytes, of the content file.</p> <p>This property cannot record a size greater than 2GB. If the file is large, examine the full_content_size property to obtain the full size.</p> <p>The property is not set if the content is stored in external storage.</p>
content_state	integer	S	Content state.

Property	Datatype	Single or repeating	Description
data_ticket	integer	S	<p>Value used internally to retrieve the content.</p> <p>An actual data ticket is created for content stored in file stores and blob stores.</p> <p>If the content is stored in turbo storage, in the content object, data_ticket is 0.</p> <p>If the content is stored in turbo storage and is too large for the <i>i_contents</i> property in the content object, the content is stored in repeating values of the <i>i_contents</i> property of a subcontent object.</p> <p>The data_ticket value in the content object contains the number of values in that properties used to store the content.</p> <p>If the content is stored in external storage, data_ticket is 0.</p> <p>If the content is stored in a distributed storage area component, the data ticket value identifies a <i>dmi_replica_record</i> object for the content.</p> <p>If the content is stored in content-addressed storage, the data_ticket is a number used internally only and</p>

Property	Datatype	Single or repeating	Description
			has no external meaning.
encoding	string(10)	S	The encoding for the format, if any.
format	ID	S	Object ID of the format object describing this content's format.
full_content_size	double	S	Size, in bytes, of the content file. The property is not set if the content is stored in external storage.
full_format	string(64)	S	Full format specification for the content.
fulltext_index	ID	R	obsolete

Property	Datatype	Single or repeating	Description
i_parked_state	integer	S	<p>Specifies the inbound transfer status of the content. Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning the content is in the storage area. • 1, meaning the content is temporarily residing on a Branch Office Caching Services server. • 2, meaning the content is temporarily residing on an Accelerated Content Services server. • 3, meaning that the content is not available because this content object was replicated from a repository whose content is not yet available. That is, the source content has a non-zero value in i_parked_state.
i_partition	integer	S	Specifies the partition where the object resides.
index_format	ID	S	obsolete
index_formats	ID	R	obsolete
index_operations	integer	R	obsolete
index_pages	integer	R	obsolete
index_parent	ID	S	obsolete
index_parents	ID	R	obsolete
index_set_times	DATE	R	obsolete
index_subtypes	string(27)	R	obsolete

Property	Datatype	Single or repeating	Description
is_archived	Boolean	S	Indicates whether the content has been archived.
is_offline	Boolean	S	Indicates whether the content is in the storage area.
i_contents	string() The size of the string is the maximum for the underlying database.	S	If the content is stored in turbo storage, this property contains the actual content file. If the content is too large for this property, the content is stored in a dmi_subcontent object and this property is unused. If the content is stored in content-addressed storage, this property contains the content address. If the content is stored in external storage, this property contains the token used to retrieve the content. The property cannot be selected using the API or DQL.
i_encoding	string(10)	R	Contains format information used internally in the management of distributed repositories.
i_format	ID	R	Contains the information from the page and format properties.

Property	Datatype	Single or repeating	Description
i_full_format	string(64)	R	Contains format information used internally in the management of distributed repositories.
i_index_format	ID	R	obsolete
i_other_contents	string() The size of the string is the maximum for the underlying database.	S	(Not implemented yet in release 6.6) The content address of the resource fork.
i_px	integer	R	Contains format information used internally in the management of distributed repositories.
i_py	integer	R	Contains format information used internally in the management of distributed repositories.
i_pz	integer	R	Contains format information used internally in the management of distributed repositories.
i_rendition	integer	R	Contains the information in the rendition property. Used internally in the management of distributed repositories.
loss	integer	S	The transformation loss for the format, if any.
other_file_size	double	S	Reserved for future use.
other_ticket	integer	S	Value used internally to retrieve the content.

Property	Datatype	Single or repeating	Description
page	integer	R	Position of the content in each of the objects that contain it.
page_modifier	string(16)	R	User-defined string to disambiguate renditions having the same format that are associated with a particular content page of a document.
parent_count	integer	S	Total number of objects that contain this content.
parent_id	ID	R	Object IDs of the objects that contain the content represented by this content object.
r_content_hash	string(256)	S	Hashed value of the associated content file. This is only set if content is stored in a file store storage area and the storage area's content_hash_mode property is set to 1.
rendition	integer	S	<p>Provides information about a rendition. Valid values are:</p> <ul style="list-style-type: none"> • 0, for original content. • 1, for a rendition generated by the server. • 2, for a rendition generated by the client. • 3, meaning keep the rendition when the content with which it is associated is updated or removed from the document or repository.

Property	Datatype	Single or repeating	Description
resolution	integer	S	The resolution specification for the content's format.
set_client	string(64)	S	Name of the client machine on which the setfile was executed.
set_file	string(255)	S	Source file on the client machine that contained the content.
set_time	DATE	S	The initial value is the time at which the Setfile was executed. The time value is the time on the server machine.
storage_id	ID	S	Object ID of the store object representing the storage area that contains the content.
transform_path	string(32)	S	The transformation path for this format.
update_count	integer	R	Number of the update operation in which this content will be indexed.
x_range	integer	S	The format's range along the x axis.
y_range	integer	S	The format's range along the y axis.
z_range	integer	S	The format's range along the z axis.

3.50 Cryptographic key

Stores a private cryptographic key.

3.50.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_cryptographic_key
- Object type tag: 08

A cryptographic key object stores a private cryptographic key. The cryptographic key object is created automatically by Documentum CM Server. It is used by Documentum CM Server to encrypt instructions regarding content file availability for an Accelerated Content Services server or Branch Office Caching Services server. There is only one cryptographic key object in a repository. The permissions on this object give Delete permission to the owner and the members of the dm_superusers group.



Caution

These objects are for internal use only. Do not modify, remove, or add these objects.

3.50.2 Properties

The table describes the object properties.

Table 3-50: Cryptographic key type properties

Property	Datatype	Single or repeating	Description
key_identifier	string(40)	S	Base 64-encoded SHA1 digest of the DER private key.
key_type	integer	S	Specifies what the key used for by Documentum CM Server. The only valid value is: 1, meaning used for Accelerated Content Services encryption
key_value	string(1000)	S	Base 64-encoded and DBK-encrypted DER key.

Property	Datatype	Single or repeating	Description
public_key_identifier	string(40)	S	Base 64-encoded SHA1 digest of the DER public key value in the key_identifier property of the dm_public_key_certificate object.

3.51 DD attr info

Contains the published data dictionary information for a property.

3.51.1 Description

- Supertype: DD Common Info
- Subtypes: None
- Internal name: dmi_dd_attr_info
- Object type tag: 6a

A dd attr info object contains the published data dictionary information for a property. You cannot create dd attr info objects directly. They are created or modified as needed when data dictionary information is published.

With one exception (attr_name), each property defined for the type has a corresponding property in the type named i_property_name. The i_properties are used internally by Documentum CM Server.

3.51.2 Properties

The table lists the basic properties of a dd attr info object.

Table 3-51: DD attr info type properties

Property	Datatype	Single or repeating	Description
allowed_search_ops	integer	R	<p>The search operators available for the property. Valid values depend on the property's data type.</p> <p>For String:</p> <ul style="list-style-type: none"> • 1, meaning = • 2, meaning <> • 3, meaning > • 4, meaning < • 5, meaning >= • 6, meaning <= • 7, meaning begins with • 8, meaning contains • 9, meaning does not contain • 10, meaning ends with • 11, meaning in • 12, meaning not in • 14, meaning is null • 15, meaning is not null

Property	Datatype	Single or repeating	Description
allowed_search_ops			<p>For Integer:</p> <ul style="list-style-type: none">• 1, meaning =• 2, meaning \diamond• 3, meaning >• 4, meaning <• 5, meaning \geq• 6, meaning \leq• 11, meaning in• 12, meaning not in• 13, meaning between• 14, meaning is null• 15, meaning is not null <p>For Date:</p> <ul style="list-style-type: none">• 1, meaning =• 2, meaning \diamond• 3, meaning >• 4, meaning <• 5, meaning \geq• 6, meaning \leq• 13, meaning between• 14, meaning is null• 15, meaning is not null <p>For Boolean and ID:</p> <ul style="list-style-type: none">• 1, meaning =• 2, meaning \diamond• 14, meaning is null• 15, meaning is not null

Property	Datatype	Single or repeating	Description
attr_name	string(32)	S	Name of the property described by the dd attr info object. This property has no corresponding i_ property in the type.
category_name	string(64)	S	User-defined.
computed_dep_usr	Boolean	S	Indicates whether the values in computed_expr_dep were set by the user or calculated. TRUE means the values are user defined; FALSE means they are calculated.
computed_expr_dep	string(32)	R	List of properties on which the computed expression referenced in cond_computed_expr depends.
cond_computed_expr	ID	R	Object ID of the dm_cond_id_expr object that contains the conditional computed expressions and the associated object IDs.
cond_value_assist	ID	S	Object ID of the cond ID expr object that contains the conditional expressions for the property's value assistance.
def_value_length	integer	S	Length of the property's default value. A zero indicates that the value is unspecified.
default_expr_builtin	integer	R	Used internally.
default_expr_kind	integer	R	Used internally.
default_expr_value	string(255)	R	Used internally to optimize default value handling.

Property	Datatype	Single or repeating	Description
default_search_arg	integer	S	Default value to display in conjunction with the default search operator.
default_search_op	string(255)	S	Default search operator to display to users when they search on the property.
default_value	ID	R	Object ID of an expression object that resolves to the default value for the property. For single-valued properties, only one value can be defined. For repeating properties, multiple values can be defined.
domain_length	integer	S	For string properties, the maximum length of a value.
domain_type	integer	S	Specifies the datatype of the property. Valid values are: <ul style="list-style-type: none"> • 0, Boolean • 1, Integer • 2, String • 3, ID • 4, Time/Date • 5, Double
format_pattern	string(64)	S	For date properties, the pattern used to interpret values for the property.
format_pattern_tag	integer	S	Currently unused. This value is always 1.

Property	Datatype	Single or repeating	Description
ignore_immutable	Boolean	S	Indicates whether the property is changeable even if the containing object is immutable. This setting is effective only for objects of type dm_sysobject or its subtypes.
is_hidden	Boolean	S	For use by client applications.
is_required	Boolean	S	Indicates whether users must provide a value for the property.
map_data_string	string(128)	R	List of possible data values for the property.
map_description	string(255)	R	Descriptions of the data values at the corresponding index levels in map_data_string.
map_display_string	string(128)	R	The character string to display for the data value at the corresponding index level in map_data_string.
not_null	Boolean	S	Indicates whether the property has a NOT NULL constraint defined for it.
not_null_enf	integer	S	Indicates who is responsible for enforcing a NOT NULL constraint. Valid values are the same as for foreign_key_enfs.
not_null_msg	string(255)	S	Error message to display when the NOT NULL constraint is violated.

Property	Datatype	Single or repeating	Description
read_only	Boolean	S	Indicates whether the property is read only.
reference_kind	integer	S	Used internally by Collaboration Services.
super_domain_id	ID	S	Object ID of the domain from which the domain of this property is derived, if any.
super_domain_name	string(32)	S	Name of the domain from which the domain of this property is derived. If blank, the domain is the built-in domain indicated by domain_type.
value_assist_dep	string(32)	R	Properties on which the value assistance expressions depend.
value_assist_dep_usr	Boolean	S	Indicates whether the user defined the list of values in value_assist_dep. TRUE means the user defined the values; FALSE means the values were calculated.

3.52 DD common info

Contains the published data dictionary information that an object type and property have in common.

3.52.1 Description

- Supertype: Persistent Object
- Subtypes: DD Attr Info; DD Type Info
- Internal name: dmi_dd_common_info
- Object type tag: 68

A dd common info object contains the published data dictionary information that an object type and property have in common. You cannot create a dd common info object directly. They are created as needed when data dictionary information is published.

With four exceptions, each property listed in the table, “[DD common info type properties](#)” on page 169, has a corresponding property named *i_property_name* in the object type definition. The *i*_properties are used internally by OpenText Documentum CM products. The four exceptions are the properties that identify the object type or property with which the dd common info object is associated. These four properties are:

- business_policy_id
- nls_key
- state_name
- type_name

3.52.2 Properties

The table lists the properties defined for the object type.

Table 3-52: DD common info type properties

Property	Datatype	Single or repeating	Description
business_policy_id	ID	S	<p>Object ID of a dm_policy object.</p> <p>This property has a value only if the information in the dd common info object is associated with a particular lifecycle and state.</p> <p>This property has no corresponding <i>i</i>_property in the type.</p>
comment_text	string(255)	S	User-defined.

Property	Datatype	Single or repeating	Description
constraint_dep_usr	Boolean	S	TRUE indicates that the values in val_constraint_dep are user-defined. FALSE means that the values are computed by Documentum CM Server.
foreign_keys	ID	R	Object IDs of any foreign key objects associated with the property.
foreign_key_enfs	integer	R	Indicates who is responsible for enforcing the corresponding foreign key. Valid values are: 1, Enforcement disabled 2, Enforcement by application
foreign_key_msgs	string(255)	R	Error message to display when the corresponding foreign key constraint is violated.
help_text	string(255)	S	Help text to use for the property.
i_dd_flags	integer	R	Reserved for future use.
ignore_constraints	Boolean	S	TRUE indicates that constraints inherited from parent types are ignored. FALSE means the inherited constraints are enforced. The default is FALSE.
is_searchable	Boolean	S	Indicates whether the property is searchable. The default is TRUE, meaning the property can be searched.

Property	Datatype	Single or repeating	Description
label_text	string(64)	S	<p>Label for the property or object type.</p> <p>OpenText Documentum CM provides default labels for all properties and object types. For numerous properties, the default is defined in a data dictionary population file.</p> <p>If no default is defined in the data dictionary, the default for an object type is its name. For example, for dm_sysobject, the default is dm_sysobject. The default for a property is the name of the property.</p>
life_cycle	integer	S	<p>Indicates the current life cycle status of the type or property. Valid values are:</p> <ul style="list-style-type: none"> 1, Currently in use 2, For future use 3, Obsolete <p>The default is 1 (currently in use).</p>
nls_key	string(5)	S	<p>Locale of the information in the dd common info object.</p> <p>This property has no corresponding i_ property in the type.</p>

Property	Datatype	Single or repeating	Description
primary_key	ID	S	Object ID of dm_key object that contains primary key constraints for the type or property.
primary_key_enf	integer	S	Specifies who is responsible for enforcing the corresponding primary key. Valid values are: 1, Enforcement disabled 2, Enforcement by application
primary_key_msg	string(255)	S	Error message to display when the primary key is violated.
resync_needed	Boolean	S	Indicates whether there are unpublished changes to the object or property. TRUE indicates that there are unpublished changes.
state_name	string(32)	S	Name of the lifecycle state in which this data dictionary information applies. This property has a value only if the information in the dd common info object is associated with a particular lifecycle and state. This property has no corresponding i_ property in the type.

Property	Datatype	Single or repeating	Description
type_name	string(32)	S	Name of the object type with which the information in the dd common info object is associated. This property has no corresponding <i>i_</i> property in the type.
unique_keys	ID	R	Object IDs of the key objects representing any unique keys that apply to the type or property. The unique keys defined for the type or property are listed first, followed by those inherited from the supertype or super domain.
unique_key_enfs	integer	R	Indicates who is responsible for enforcing the corresponding unique key. Valid values are the same as for foreign_key_enfs.
unique_key_msgs	string(255)	R	Error message to display when the corresponding unique key is violated.
val_constraint	ID	R	Object IDs of the expression objects representing any check constraints that apply to the property.
val_constraint_dep	string(32)	R	Names of the properties referenced by the constraint at the corresponding index level in val_constraint.

Property	Datatype	Single or repeating	Description
val_constraint_enf	integer	R	Indicates who is responsible for enforcing the corresponding validation constraint. Valid values are the same as for foreign_key_enfs.
val_constraint_msg	string(255)	R	Error message to display when the corresponding validation constraint is violated.

3.53 DD info

Contains data dictionary information for an object type or property that is not dependent on locale.

3.53.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dm_dd_info
- Object type tag: 4e

A dd info object contains data dictionary information for an object type or object property that is not dependent on locale. Documentum CM Server creates and manages dd info objects. Properties in dd info objects are set when users add or change data dictionary information.

3.53.2 Properties

The table, “[DD info properties for object types and object properties](#)” on page 174, lists the properties defined for dd info that are applicable to both object types and properties. The table, “[DD info properties for object types](#)” on page 177, lists the properties that apply only to object types. The table, “[DD info properties for object properties](#)” on page 178, lists the dd info properties that apply only to properties.

Table 3-53: DD info properties for object types and object properties

Property	Datatype	Single or repeating	Description
auditable_appevents	string(64)	R	The application-defined events that can be audited.

Property	Datatype	Single or repeating	Description
auditable_sysevents	string(64)	R	The system-defined events that can be audited.
foreign_keys	ID	R	Object IDs of the foreign key objects for the type or property.
foreign_key_enfs	integer	R	Indicates how the corresponding foreign keys are enforced. Valid values are: <ul style="list-style-type: none"> • 1, Enforcement disabled • 2, Enforcement by application
fulltext_support	integer	S	Indicates how full-text indexing is supported for the object type. Valid values are: <ul style="list-style-type: none"> • 0, no support • 2, full support <p> Note: Currently, this is only used by lightweight object subtypes.</p>
ignore_constraints	integer	S	Indicates whether the validation, key, and NOT NULL constraints defined for the type or are ignored. Valid values are: <ul style="list-style-type: none"> • 0, Follow the constraints • 1, Ignore the constraints • -1, Inherit the value from the corresponding property in the type or type's supertype

Property	Datatype	Single or repeating	Description
is_searchable	integer	S	<p>For client use. Indicates whether the type or type should appear in pick lists for Find dialogs. Valid values are:</p> <ul style="list-style-type: none"> • 0, FALSE • 1, TRUE • -1, Inherit the value from the corresponding property in the type or supertype of type
life_cycle	integer	S	<p>Indicates the state of the object type or in the repository. Valid values are:</p> <ul style="list-style-type: none"> • 1, Currently in use • 2, For future use • 3, Obsolete
unique_keys	ID	R	Object IDs of the key objects that define unique keys for this type or property.
unique_key_enfs	integer	R	Indicates how the corresponding unique keys are enforced. Valid values are the same as for foreign_key_enfs.
val_constraint	ID	R	Object IDs of the expression objects corresponding to the constraints applied to the type or property.
val_constraint_dep	string(32)	R	The properties on which the expression validation constraints depend.

Property	Datatype	Single or repeating	Description
val_constraint_enf	integer	R	Indicates how the corresponding validation constraints are enforced. Valid values are the same as for foreign_key_enfs.

Table 3-54: DD info properties for object types

Property	Datatype	Single or repeating	Description
comp_classifier	string(128)	R	Qualified components that can be called for this type.
default_policy_id	ID	S	Chronicle ID of the default policy object for the type.
icon_index	integer	S	The index that locates the icon of the type in the icon resource file.
policy_ver_label	string(32)	S	Version label that identifies which version of the default policy object to use for the type.
primary_key	ID	S	Object ID of the primary key for the type. The primary key must be one of the unique keys defined in unique_key.
primary_key_enf	integer	S	Indicates how the primary key is enforced. Valid values are the same as for foreign_key_enfs.
qual_comp_id	ID	R	Object IDs corresponding to the qualified components identified in comp_classifier.

Table 3-55: DD info properties for object properties

Property	Datatype	Single or repeating	Description
allowed_search_ops	integer	R	A list of integers representing the valid search operators for this property. Valid values are: <ul style="list-style-type: none">• 1, for =• 2, for <>• 3, for >• 4, for <• 5, for >=• 6, for <=• 7, for begins with• 8, for contains• 9, for does not contain• 10, for ends with• 11, for in• 12, for not in• 13, for between• 14, for is null• 15, for is not null• 16, for not
computed_expr_dep	string(32)	R	Currently unused.
cond_computed_expr	ID	S	Currently unused.
cond_value_assist	ID	S	Object ID of the cond id expr object that contains the conditional value assistance.
default_search_arg	string(255)	S	The default value to use in conjunction with the default search operator (defined in default_search_op).

Property	Datatype	Single or repeating	Description
default_search_op	integer	S	The default search operator for the property. The operator specified must be in allowed_search_ops.
default_value	ID	R	<p>Object IDs of the expression objects representing the property's default values.</p> <p>For single-valued properties, default_value[0] must be the object ID of a dm_expression object corresponding to a literal of the appropriate datatype for the property or the NULL value appropriate for the type (for example, NULLSTRING).</p> <p>For repeating properties, multiple expression object IDs are allowed, representing multiple default values for the property. However, object IDs representing expressions that evaluate to NULL are not allowed.</p>

Property	Datatype	Single or repeating	Description
ignore.Immutable	integer	S	<p>Indicates whether the value in <code>r.Immutable_flag</code> controls the property's changeability. This setting affects only properties of objects of type <code>dm_sysobject</code> or <code>SysObject</code> subtypes.</p> <p>If set to 1 (TRUE), the is changeable regardless of the object's <code>r.Immutable_flag</code> setting. Valid values are:</p> <ul style="list-style-type: none"> • 0, FALSE • 1, TRUE • -1, Inherit value from super domain or type
is_hidden	integer	S	<p>Provided for use by client applications. Valid values are the same as for <code>ignore.Immutable</code>.</p>
is_required	integer	S	<p>Indicates whether a value is required for the property.</p> <p>If set to 1 (TRUE), the must have a value before saving the object. Valid values are the same as for <code>ignore.Immutable</code>.</p>

Property	Datatype	Single or repeating	Description
not_null	integer	S	Indicates whether the has the NOT NULL constraint defined for it. If set to 1 (TRUE), the NOT NULL constraint is specified for the property. Valid values are the same as for ignore_immutable.
not_null_enf	integer	S	Indicates how the NOT NULL constraint is enforced. Valid values are the same as for the foreign_key_enfs.
parent_id	ID	S	Object ID of the aggr domain object that references the dd info object.
read_only	integer	S	Indicates whether users can read and write the property or only read it. If set to 1 (TRUE), users can only read this property, they cannot write to it. Valid values are the same as for ignore_immutable.
reference_kind	integer	S	Used internally by Collaboration Services
value_assist_dep	string(32)	R	Properties on which the value assistance depends.

3.54 DD type info

Contains the published data dictionary information for an object type.

3.54.1 Description

- Supertype: DD Common Info
- Subtypes: None
- Internal name: dmi_dd_type_info
- Object type tag: 69

A dd type info object contains the published data dictionary information for an object type. You cannot create or modify dd type info objects directly. They are created or modified as needed when data dictionary information is published.

3.54.2 Properties

The table lists the properties of the object type.

Table 3-56: DD type info type properties

Property	Datatype	Single or repeating	Description
attr_domain_name	string(32)	R	Currently unused.
attr_domain_id	ID	R	Currently unused.
auditable_appevents	string(64)	R	The application-defined events that can be audited.
auditable_sysevents	string(64)	R	The system-defined events that can be audited.
comp_classifier	string(128)	R	The classifiers for the qualified components that can be executed against instances of the type.
default_policy_id	ID	S	Object ID of the dm_policy object representing the default lifecycle for the type.
i_attr_domain_id	integer	S	For internal use.
i_attr_domain_name	integer	S	For internal use.
i_comp_classifier	integer	S	For internal use.
i_default_policy_id	Boolean	S	For internal use.

Property	Datatype	Single or repeating	Description
i_icon_index	Boolean	S	For internal use.
i_policy_version_label	Boolean	S	For internal use.
i_qual_comp_id	integer	S	For internal use.
icon_index	integer	S	The index that locates the icon for the type in the icon resource file.
map_data_string	string(128)	R	List of possible data values for the property.
map_description	string(255)	R	Descriptions of the data values at the corresponding index levels in map_data_string.
map_display_string	string(128)	R	The character string to display for the data value at the corresponding index level in map_data_string.
policy_ver_label	string(32)	S	Version label of the object identified in default_policy_id.
			 Note: This value is provided by the user, not derived from the object by Documentum CM Server.
qual_comp_id	ID	R	Object IDs of the qual comp objects representing the component routines that can be executed against instances of the type.
r_has_check	Boolean	R	For internal use.
r_has_constraint	Boolean	R	For internal use.
r_has_default	Boolean	R	For internal use.
r_has_dependency	Boolean	R	For internal use.

Property	Datatype	Single or repeating	Description
r_has_foreign_key	Boolean	R	For internal use.
r_has_ignore_immutable	Boolean	R	For internal use.
r_has_not_null	Boolean	R	For internal use.
r_has_primary_key	Boolean	R	For internal use.
r_has_unique_key	Boolean	R	For internal use.
r_has_value_assist	Boolean	R	For internal use.
scope_config	ID	R	Object IDs of the scope config object for the object type.

3.55 Display config

Defines the display configuration for a group of properties.

3.55.1 Description

- Supertype: None
- Subtypes: None
- Internal name: dm_display_config
- Object type tag: 6b

A display config object defines the display configuration for a group of properties. Display config objects are referenced by scope config objects, which define the context within which the display configuration is used. Both display config objects and scope config objects are used by client applications. Documentum CM Server does not use these objects.

3.55.2 Properties

The table lists the properties defined for the type.

Table 3-57: Display config type properties

Property	Datatype	Single or repeating	Description
attribute_display_hint	integer	R	<p>Controls the display of properties in the user interface. These values are interpreted by client application, not Documentum CM Server.</p> <p>Values used by Desktop Client and Webtop are:</p> <ul style="list-style-type: none"> • 0, meaning do not display a separator before a property • 1, meaning display a separator before a property <p>Values used by Webtop only are:</p> <ul style="list-style-type: none"> • 2, meaning display the property if the user requests it and do not use a separator • 3, meaning display the property if the user requests it and use a separator <p>The hint at a particular index position is applied to the property named at the corresponding index position in attribute_name.</p>

Property	Datatype	Single or repeating	Description
attribute_name	string(40)	R	List of properties names. The properties must be defined for the object type identified in property source. The order in which the properties are listed determines the order in which they are displayed in the client application.
attribute_source	string(27)	S	The object type for which the properties listed in attribute_name are defined.
fixed_display	Boolean	S	Indicates whether properties can be added to attribute_name or display_hints changed. T means that the property list and hint cannot be changed. F means that the property list and hint can be changed. The default is F.
i_config_identifier	string(20)	S	Uniquely identifies a display config object across repositories. The format of the value is dm_r_object_id, where r_object_id is the object ID of the display config object. This is used internally to manage load operations.

Property	Datatype	Single or repeating	Description
object_name	string(64)	S	<p>Name of the display config object. This must be unique within the scope in which the display config object is used.</p> <p> Note: Uniqueness is enforced by the client applications, not Documentum CM Server.</p>

3.56 Distributed store

Contains information about a storage area.

3.56.1 Description

- Supertype: Store
- Subtypes: None
- Internal name: dm_distributedstore
- Object type tag: 2c

A distributed store object contains information about a storage area. A distributed storage area points to component storage areas and is used to implement a repository that has distributed content. All but three of its properties are inherited from its supertype, dm_store.



Note: For more information about distributed storage areas, see *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY250400-IGD)*.

3.56.2 Properties

The table lists the properties defined for the distributed store type.

Table 3-58: Distributed store type properties

Property	Datatype	Single or repeating	Description
change_record_id	ID	S	Used internally to manage structural changes.
epoch_number	integer	S	Records a count of how many structural changes (adding or removing component areas) have been made to the distributed storage area.
only_fetch_close	Boolean	S	Indicates if a server only directly fetches files from component storage areas in the distributed storage area that are defined as close to the server. Typically set to false for distributed stores that use shared content. Typically set to true for distributed stores that replicate content. When set to true, the surrogate get method replicates the content to the local store.

3.57 Messaging Service config

Records configuration information about a Messaging Service server.

3.57.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_dms_config
- Object type tag: 08

A dms config object records the configuration of a Messaging Service server. A Messaging Service config object is stored in the global registry repository. There can be only one Messaging Service config object in each global registry in the installation. Use Documentum Administrator to create and manage a Messaging Service config object.

3.57.2 Properties

The table lists the properties defined for the type.

Table 3-59: Messaging Service config type properties

Property	Datatype	Single or repeating	Description
dms_capability	integer	S	<p>Defines the capabilities of the Messaging Service server. Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning disabled • 1, meaning enabled <p>The default is 1.</p>
message_consume_url	string(240)	S	The URL used by a Branch Office Caching Services to retrieve messages from a Messaging Service server.
message_post_url	string(240)	S	The URL used by Foundation Java API to post messages to the Messaging Service.

Property	Datatype	Single or repeating	Description
server_major_version	string(16)	S	Indicates the major version of the Messaging Service server. For a Messaging Service 6.0 server, this must be set to 1.
server_major_version	string(16)	S	Indicates the minor version level of the Messaging Service server. For a Messaging Service 6.0 server, this must be set to 0.

3.58 Docbase config

Contains configuration information about a repository.

3.58.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_docbase_config
- Object type tag: 3c

A docbase config object contains configuration information about a repository. Each repository must have a single docbase config object whose object name matches the name of the repository.

3.58.2 Properties

The table lists the properties defined for the type.

Table 3-60: Docbase config type properties

Property	Datatype	Single or repeating	Description
a_bpaction_run_as	string(255)	S	<p>Defines which user account is used to run lifecycle (business policy) actions. Valid values are:</p> <ul style="list-style-type: none"> • session_user (default) • superuser • lifecycle_owner • <i>user name</i> <p>The first three values are keywords that must be entered as shown. For the last, <i>user name</i>, specify the user name of a repository user.</p>
approved_clients_only	Boolean	S	Indicates whether clients connecting to this repository must have an entry in the Client Rights database.
audit_old_values	Boolean	S	Indicates whether to include old as well as new values of audited properties in the audit trail entry. TRUE means to include the old values. FALSE means to exclude them. The default is TRUE.
auth_deactivation_interval	integer	S	<p>Length of time between a user's account deactivation and automatic reactivation. If this is 0, the account is not automatically reactivated.</p> <p>The value is specified in minutes.</p> <p>The default is 0.</p>

Property	Datatype	Single or repeating	Description
auth_failure_interval	integer	S	<p>Length of time, in minutes, in which consecutive failed login authorizations will cause a user's account to be deactivated. The number of failed attempts that must occur within the interval to trigger deactivation is determined by the max_auth_attempts property.</p> <p>The default is 0, meaning that deactivation always occurs when the maximum number of consecutive failed login attempts is reached, regardless of how long that takes.</p>
auth_protocol	string(32)	S	<p>On Windows platforms, if set to domain_required, it indicates that the repository is running in domain-required mode. If the repository is not using domain-required mode, this is blank.</p> <p>On Linux platforms, if you are authenticating users against a Windows domain, set this to unix_domain_used. Otherwise, this property is blank for repositories running on a Linux platform.</p>

Property	Datatype	Single or repeating	Description
check_client_version	Boolean	S	T means that the repository servers will not accept connections from clients older than the version level specified in the oldest_client_version property. F means that the servers accept connections from any client version. The default is F.
client_pcaching_change	integer	S	Controls persistent client cache flushing. Incrementing this value forces clients to flush all persistent caches on start up.
client_pcaching_disabled	Boolean	S	T (TRUE) disables persistent client caching for sessions with the repository. The default is F (caching is allowed).
dd_locales	string(5)	R	Data dictionary locales recognized by the server. Setting this property requires you to execute a Reinit method on the server to make the change visible.

Property	Datatype	Single or repeating	Description
default_app_permit	integer	S	<p>Default user permission level for application-controlled objects accessed through an application that doesn't own the object. Values are:</p> <ul style="list-style-type: none"> • 2, Browse permission • 3, Read permission • 4, Relate permission • 5, Version permission • 6, Write permission • 7, Delete permission <p>The default value is 3, Read permission.</p>
dir_user_sync_on_demand	Boolean	S	<p>Controls whether directory users can be synchronized on demand in the repository. T means that such users can be synchronized on demand. F means that they cannot be synchronized on demand.</p> <p>The default value is F.</p> <p>This property is deprecated.</p>
docbase_roles	string(32)	R	Used internally.

Property	Datatype	Single or repeating	Description
dormancy_status	string(32)	S	<p>Specifies whether the repository is active or dormant.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • ACTIVE - The repository is active. Each of the repository's Documentum CM Servers determines its own dormant status. • DORMANT - The repository is dormant. The dormant status of all of the repository's Documentum CM Servers is either DORMANT or DORMANCY_REQUESTED. <p>When each of the repository's Documentum CM Servers is restarted or re-initialized, its dormant status is set to the same value as this attribute.</p>
effective_date	date	S	Obsolete in 5.2 repositories. In pre-5.2 repositories, this is used to force persistent client cache flushes. The default value is NULLDATE.
folder_security	Boolean	S	Indicates whether the repository is running with folder security on or off. The default is T, meaning that security is turned on.
fulltext_install_locs	string(32)	R	Name of the location object that points to a fulltext installation.

Property	Datatype	Single or repeating	Description
i_crypto_key	string(255)	S	The encrypted encryption key or ID for the repository.
i_crypto_keys_expiry_date	date	R	Reserved for future use.
i_expired_crypto_keys	string(255)	R	Reserved for future use.
i_ticket_crypto_key	string(255)	S	The encrypted login ticket key or ID.
index_store	string(80)	S	Name of the RDBMS tablespace where you want to store type indexes.
login_ticket_cutoff	Date	S	Defines the earliest possible creation date for valid login tickets. Tickets issued before this date are not valid in this repository. The default value is NULLDATE, meaning there is no cutoff date for login tickets.
mac_access_protocol	string(32)	S	Specifies the type of file sharing protocols in use for Macintosh clients. Valid values are nt, double, ushare, or none. This is set to nt when a repository is configured to enable <i>other file</i> use. Use none only if you have no Macintosh clients. Double and ushare are meaningful only to Macintosh clients.

Property	Datatype	Single or repeating	Description
macl_security_disabled	Boolean	S	If T (TRUE), Documentum CM Server enforces only the AccessPermit and ExtendedPermit entries in an ACL. The default value is T (TRUE).
max_auth_attempt	integer	S	Maximum number of unsuccessful login attempts allowed. The default is 0, which means the feature is disabled.
minimum_owner_permit	integer	S	Integer value specifying the minimum permission value for an object owner. Must be between 0 (NULL) and 7 (DELETE). However, 1 (NONE) is not an allowed value. The default value will be 0 (NULL). When 0 is specified, the server will give object owners the same minimum permit they have prior to the introduction of this setting: 3 (COMMON).

Property	Datatype	Single or repeating	Description
minimum_owner_xp ermit	integer	S	<p>Comma-delimited list of the minimum extended permissions that an object owner can have. The acceptable values are:</p> <ul style="list-style-type: none"> • execute_proc • change_location • change_state • change_permit • change_owner • delete_object • change_folder_li ks <p>The default value for this field is an empty string (in other words, object owners have no minimum extended permissions).</p>
object_name	string(255)	S	<p>Contains the name of the repository. This property is inherited from dm_sysobject.</p> <p>A repository name must be all ASCII characters and <=32 characters in length. The name docu is reserved for OpenText Documentum CM.</p>

Property	Datatype	Single or repeating	Description
offline_checkin_flag	integer	S	<p>Used by some OpenText Documentum CM clients to determine whether a client dialog box is used to check in content or the user's local check-in setting takes precedence. Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning use a client dialog box for check ins • 1, meaning allow the user's local check-in setting to take precedence <p>The default value is 0.</p>
offline_sync_level	integer	S	<p>Specifies the level of repository access for offline synchronization. This is used primarily by some OpenText Documentum CM clients. Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning none • 1, meaning one-way access • 2, meaning role-based access <p>The default value is 0.</p> <p> Note: Refer to the client documentation for information about the actual use of this property and role-based synchronization.</p>

Property	Datatype	Single or repeating	Description
oldest_client_version	string(32)	S	<p>Version number of the oldest OpenText Documentum CM client that will access this repository.</p> <p>This must be set manually. It is used by the Foundation Java API to determine how to store chunked XML documents. If check_client_version is set to T, then this value is also used to identify the oldest client version level that may connect to the repository.</p>
r_address_partitions	integer	S	This is no longer used.
r_dbms_name	string(32)	S	<p>Name of the RDBMS.</p> <p>Valid values on Windows platforms are: Oracle, SQL Server, and Postgres.</p> <p>Valid values on Linux platforms is: Oracle and Postgres.</p>
r_docbase_id	integer	S	The repository ID as a decimal value.

Property	Datatype	Single or repeating	Description
r_crypto_keystore	string(32)	S	<p>Where encryption keys are managed. This is set when the repository is created and cannot be changed. Possible values are:</p> <ul style="list-style-type: none"> • Local—means the keys are stored in the database • Remote—means the keys are stored in a remote key management server and only the key IDs are stored in the database
r_crypto_mode	string(64)	S	<p>Lists the symmetric key encryption algorithm, the asymmetric key encryption algorithm, and the hash algorithm used in the repository. The string consists of the abbreviations for each algorithm concatenated using an underscore (_). For example, a string value of:</p> <ul style="list-style-type: none"> • AES128_RSA2048_SHA3_384, indicates that AES128 is the symmetric key algorithm • RSA2048 is the asymmetric key algorithm • SHA3_384 is the hash algorithm used by the repository
r_ending_partition	integer	S	This is no longer used.

Property	Datatype	Single or repeating	Description
r_federation_name	string(120)	S	Name of the federation to which the repository belongs, if any. The name is taken from the value of the object_name property in the dm_federation object.
r_module_mode	integer	R	Used internally.
r_module_name	string(32)	R	Used internally.
r_normal_tz	integer	S	Indicates the storage mode used for date values. 0 directs Documentum CM Server to store all date values in UTC time. A value other than 0 represents the time zone offset from UTC time used by Documentum CM Server to normalize all date values for storage. A time zone offset must be expressed as seconds. For example, if the offset is for PST (Pacific Standard Time), the value is -28800 (-8*60*60). This defaults to 0 for a new 6.0 or later repository and to an offset representing the server local time for a repository that is upgraded from a pre-6.0 release.
r_starting_partition	integer	S	This is no longer used.

Property	Datatype	Single or repeating	Description
r_storage_mode	integer	S	Used internally.
richmedia_enabled	Boolean	S	Indicates whether the Documentum CM Server for this repository can process rich-media content. This value is set to TRUE during installation of a Content Transformation Services product.
security_mode	string(32)	S	Defines the security level for the repository. Valid values are acl or none.
trust_by_default	Boolean	S	<p>Specifies whether the repository accepts login tickets and application access control tokens from other repositories that have the same login ticket key (LTK) as this repository.</p> <p>T (TRUE) means the repository accepts login tickets and tokens from other repositories that share its LTK.</p> <p>F (FALSE) means that it only accepts login tickets and tokens from other repositories sharing its LTK if they appear in the trusted_docbases list.</p> <p>The default value is F.</p>

Property	Datatype	Single or repeating	Description
trusted_docbases	string(255)	R	Name of the repositories from which login tickets and tokens are accepted. The repositories must have the same login ticket key as the repository represented by this docbase config object.
wf_package_control_enabled	Boolean	S	<p>Controls whether Documentum CM Server exposes the object names of components in workflow packages.</p> <p>F allows Documentum CM Server to set the r_component_name property in dmi_package objects. T disallows setting the r_component_name property in package objects.</p> <p>If this property is T, the setting at the workflow level, in the package_control property of the dm_process object is ignored.</p> <p>The default value is F.</p>

3.59 Docset

Represents a set of documents to be processed by Content Intelligence Services.

3.59.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dm_docset
- Object type tag: 09

A docset object represents a set of documents to be processed by Content Intelligence Services. The documents are defined by a query recorded in the docset content. The format of the content is crtext.

3.59.2 Properties

The table lists the properties defined for the type.

Table 3-61: Docset type properties

Property	Datatype	Single or repeating	Description
description	string(255)	S	User-defined description of the document set.
last_run_id	ID	S	Object ID of the docset run object last used to process this document set.
object_name	string(255)	S	Document set name.
qualifiers	string(255)	R	List of conditions that documents must meet for inclusion in the docset.
queue_bound	Boolean	S	Controls whether documents placed on the processing queue can be considered for inclusion in this docset. Setting this to T allows documents on the processing queue to be considered for inclusion in this docset.

Property	Datatype	Single or repeating	Description
query_type	string(32)	S	Specifies the type of query. The only valid value is dq1.

3.60 Docset run

Represents an executable collection of documents.

3.60.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dm_docset_run
- Object type tag: 09

A docset run object represents an executable collection of documents. Docset run objects are used by Content Intelligence Services.

3.60.2 Properties

The table lists the properties defined for the type and those inherited properties that have a meaning specific to this type.

Table 3-62: Docset run type properties

Property	Datatype	Single or repeating	Description
docset_id	ID	S	Object ID of the document set associated with the processing run.
estimated_count	integer	S	Estimated number of documents to process in the run.
object_name	string(255)	S	Name of the docset run.
processed_count	integer	S	Actual number of documents processed by the run.
run_interval	integer	S	Not currently used.
run_mode	integer	S	Not currently used.

Property	Datatype	Single or repeating	Description
run_now	Boolean	S	T (TRUE) invokes processing immediately. The default is F (FALSE).
run_owner	string(255)	S	Name of the category manager who started the processing.
run_status	string(32)	S	Status of the processing run. Valid values are: <ul style="list-style-type: none"> • not_started • started • scheduled • completed • failed
run_type	string(32)	S	Indicates whether this is a test run or an actual run. Valid values are: <ul style="list-style-type: none"> • test • active
start_time	Date	S	Not currently used.
taxonomy_id	ID	R	Object IDs of all taxonomies included in the processing. Currently, only one taxonomy in a run is supported.
title	string(255)	S	Description of status state. The description may be user or system-defined.

3.61 Document

Contains information about a document.

3.61.1 Description

- Supertype: SysObject
- Subtypes: Too many to list.
- Internal name: dm_document
- Object type tag: 09

Documents are the objects most commonly manipulated by users in the OpenText Documentum CM system. Documents can be simple documents or virtual documents. In a simple document, the content generally seen by a user is in one or more content files associated with the document. A virtual document is a document that is composed of components that are either simple documents or other virtual documents, and the content that users see is the content files associated with these components.

A virtual document can also have associated content files. That is, in addition to its components, a virtual document can have an associated content file (or files). However, when you open a virtual document for viewing or editing through WorkSpace, WorkSpace opens the component.

All content files associated directly with a document must have the same file format. The components of virtual documents can have a mixture of formats.

3.61.2 Properties

Documents inherit all of their properties from their supertype, the SysObject type.

3.62 Domain

Describes the properties of a property's domain.

3.62.1 Description

- Supertype: Persistent Object
- Subtypes: Aggr Domain
- Internal name: dm_domain
- Object type tag: 50

A domain object describes the properties of a property's domain. The information contained in a domain object is part of the data dictionary. The domain information includes the datatype of the property, length (if a string type), localized label text, and any expressions used to populate the property. Domain objects are created and managed by the server and cannot be created by users.

3.62.2 Properties

The table lists the properties defined for the type.

Table 3-63: Domain type properties

Property	Datatype	Single or repeating	Description
dd_info	ID	S	Object ID of the dd info object containing the information of domain that does not depend on locale.
domain_length	integer	S	For string domains, the maximum length of the domain.
domain_name	string(32)	S	Currently unused.
domain_type	integer	S	Indicates the datatype of the domain. Valid values are: <ul style="list-style-type: none"> • 0, Boolean • 1, Integer • 2, String • 3, ID • 4, Time/Date • 5, Double
nls_dd_info	ID	R	Object IDs of the nls dd info objects containing the information of the domain that is dependent on locale.
nls_default	integer	S	Currently unused.

Property	Datatype	Single or repeating	Description
nls_keys	string(64)	R	<p>Specifies the locales for which localized information is available. The values are the standard locale codes; for example, en or ja.</p> <p>The localized domain information for each locale is recorded in the nls_dd_info object identified in the corresponding index position of the nls_dd_info property.</p> <p>This property is set when a locale is populated.</p>
parent_id	ID	S	<p>Object ID of the aggr domain object that references this domain in its attr_domain_id property.</p>
re_sync_dd	Boolean	S	<p>Indicates whether the domain's property values accurately reflect the values in the associated type definition.</p> <p>If TRUE, indicates that the domain may not accurately reflect the information in the type definition.</p>
super_domain_id	ID	S	<p>Object ID of the domain object from which this domain is derived, if any.</p>
super_domain_name	string(32)	S	Currently unused.

3.63 Dump object record

Contains information about an object that has been copied from a repository into an external file using the Dump utility.

3.63.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_dump_object_record
- Object type tag: 30

A dump object record object contains information about an object that has been copied from a repository into an external file using the Dump utility. The information in a dump object record object is primarily useful if the dump process is interrupted and must be restarted.

3.63.2 Properties

The table lists the properties defined for the type.

Table 3-64: Dump object record type properties

Property	Datatype	Single or repeating	Description
extra_data_dumped	Boolean	S	Used internally by the dmarchive utility.
dump_object	ID	S	Object ID of the dump record object associated with the dump procedure that generated this dump object record object.
object_id	ID	S	Object ID of the object that was copied from the repository.
version	integer	S	Version stamp of the dumped object.

3.64 Dump record

Contains information about a specific dump execution.

3.64.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dm_dump_record
- Object type tag: 2f

A dump record object contains information about a specific dump execution. The server uses this information to start a dump execution. Additionally, some of the properties contain recovery information if the dump process is interrupted and must be restarted.



Note: To start a dump process, you create a dump record object and save it. Saving a dump record object automatically starts the dump process. You must have Sysadmin or Superuser privileges to create and save a dump record object.

3.64.2 Properties

The table lists the properties defined for the type.

Table 3-65: Dump record type properties

Property	Datatype	Single or repeating	Description
dump_parameter	string(255)	R	<p>Defines parameters for the dump operation, such as the cache size or content compression. Valid values are:</p> <ul style="list-style-type: none">• compress_content =T FT directs the operation to compress the content. The default is F.• cache_size=<i>integer</i> <i>integer</i> defines the cache size in megabytes.

Property	Datatype	Single or repeating	Description
file_name	string(255)	S	Name of the external file that the system is writing the dump into.
include_content	Boolean	S	Indicates whether or not to include the actual content files in the dump file. The default is false.
predicate	string(255)	R	Predicate expression that specifies which objects of the specified types are dumped.
predicate2	string(255)	R	Extends the predicate expression defined in the predicate property at corresponding index levels. For example, predicate2[0], if defined, is concatenated at the end of predicate[0] when the dump is executed.
r_current_object_count	integer	S	Number of objects dumped. This is updated continuously throughout the process, as objects are copied to the file.
r_current_pos	integer	S	Position in the dump file at which you wrote the last entry, if file is less than 2 GB.

Property	Datatype	Single or repeating	Description
r_current_root_count	integer	S	Indicates how many of the objects qualified by the predicate have been dumped. This is constantly refreshed during the dump process as the types specified in the predicate are dumped.
r_end_time	TIME	S	Time at which the dump ended.
r_is_complete	Boolean	S	<p>Set at the completion of the dump operation. Values are:</p> <ul style="list-style-type: none"> • T (TRUE), meaning the operation completed successfully • F (FALSE), meaning the operation terminated due to an error <p> Note: The value is NULL if the dump record object was created prior to upgrading to version 5.3.</p>

Property	Datatype	Single or repeating	Description
r_is_more	Boolean	S	<p>This value is set when r_is_complete is set to T. Valid values are:</p> <ul style="list-style-type: none"> • T (TRUE), meaning the dump operation completed because it hit the objects_per_transfer limit, but there are more objects to dump • F (FALSE), meaning that the dump operation completed and there are no more object to dump. <p>The value of this property is undefined if r_is_completed is set to F (FALSE).</p>
r_root_count	integer	S	Number of objects qualified by the predicate.
r_start_time	TIME	S	Time at which the dump was started.
type	string(33)	R	Specifies the types to be dumped. Note that the predicate and type repeating properties are associated in that the predicate in position x is applied to the type in position x, the predicate in position y is applied to the type in position y, and so on.
r_current_offset	string(20)	S	Position in the dump file at which the server wrote the last entry if file is 2 GB or greater.

3.65 Email message

Stores the content of an email message.

3.65.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dm_email_message
- Object type tag: 09

An email message object stores an electronic message as content. Any user can create an email message object.



Note: This is a deprecated object type.

3.65.2 Properties

The table describes the object properties.

Table 3-66: Email message type properties

Property	Datatype	Single or repeating	Description
date_received	Date	S	The date and time the message was received.
date_published	Date	S	Date and time the message was sent.
media_type	string(32)	S	Specifies what kind of message this is. For example, possible values are email, IM (instant message), note, or calendar.
message_identifier	string(32)	S	The unique message identifier provided by the messaging system.
other_recipients	string(48)	R	Message recipients identified in the CC list.
originating_org	string(128)	S	The sender of the message.

Property	Datatype	Single or repeating	Description
recipients	string(48)	R	Message recipients identified in the To list.

3.66 Esign template

Stores a signature page template as content.

3.66.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dm_esign_template
- Object type tag: 09

An esign template object is used to store a signature page template in the repository. A default signature page template is provided with Documentum CM Server. Users can also create custom signature page templates. Each template is represented in the repository by one esign template object.

3.66.2 Properties

The table describes the object properties.

Table 3-67: Esign template type properties

Property	Datatype	Single or repeating	Description
append_to_body	Boolean	S	Specifies whether to append or pre-pend the signature page to the content. T (TRUE) means to append the content. F (FALSE) means to prepend the content. The default is F.
begin_tag	string(1)	S	Specifies the beginning delimiter for tags on the signature page template. The default is <.

Property	Datatype	Single or repeating	Description
document_type	string(32)	R	The names of the object types that can be signed using the template.
end_tag	string(1)	R	Specifies the ending delimiter for tags on the signature page template. The default is >.
max_signatures	integer	S	Specifies the maximum number of signatures that can be added to a particular version of the object. The default is 0. There is no upper limit.

3.67 Expr code

Stores generated expression source code and p-code.

3.67.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmi_expr_code
- Object type tag: 58

An expr code object stores generated expression source code and p-code. Documentum CM Server creates and manages expr code objects. Users cannot create expr code objects.

3.67.2 Properties

The table describes the object properties.

Table 3-68: Expr code type properties

Property	Datatype	Single or repeating	Description
i_type_name	string(32)	S	The name of the type for which the source code and p-code are collected.

Property	Datatype	Single or repeating	Description
parent_id	ID	S	The object ID of an aggrdomain object. Dump and load operations use this property.

3.68 Expression

Stores information needed to execute expressions defined in the data dictionary.

3.68.1 Description

- Supertype: Persistent
- Object Subtypes: Func Expr, Literal Expr, Builtin Expr
- Internal name: dm_expression
- Object type tag: 52

An expression object stores information needed to execute expressions defined in the data dictionary. Expression objects are created and managed by Documentum CM Server and cannot be created by users.

3.68.2 Properties

The table describes the object properties.

Table 3-69: Expression type properties

Property	Datatype	Single or repeating	Description
expression_name	string(32)	S	Currently unused. For the current release, this is always NULL.
expression_text	string(255)	R	The text of the expression. If the expression text is greater than 255 characters, the first 255 characters are put in expression_text[0], the second 255 in expression_text[1] and so on.

Property	Datatype	Single or repeating	Description
expression_type	integer	S	<p>The data type of the expression.</p> <p>If the expression subtype is literal expr or builtin expr, valid values are:</p> <ul style="list-style-type: none">• 0: Boolean• 1: Integer• 2: String• 3: ID• 4: Time/Date• 5: Double <p>For the func expr subtype, valid values are:</p> <ul style="list-style-type: none">• 0: Boolean• 1: Integer• 3: ID
parent_id	ID	S	The object ID of the aggr domain or policy object that references the expression object.

3.69 External file store

Represents an external file store.

3.69.1 Description

- Supertype: External Store
- Subtypes: None
- Internal name: dm_extern_file
- Object type tag: 61

An external file store object represents an external file store. The server uses external file store objects to locate content stored in external file stores.

3.69.2 Properties

The table lists the properties defined for dm_extern_file.

Table 3-70: External file store type properties

Property	Datatype	Single or repeating	Description
a_config_name	string(64)	R	Name of the server config object to use for a_location[position].
a_location	string(64)	R	Name of a location object to be used as root_name for a_config_name[position].
def_client_root	string(64)	S	Name of the client-specific location object. Default is NULL.
def_server_root	string(64)	S	Name of the server-specific location object. Default is NULL.

3.70 External free store

Represents an external storage area accessed through a user-defined content token.

3.70.1 Description

- Supertype: External Store
- Subtypes: None
- Internal name: dm_extern_free
- Object type tag: 63

The external free store object type specifies a token for content retrieval and storage that does not follow the standard for external file store or external URL store. Users can define their own token standard and means of retrieving the content associated with the token.

Users can configure the plug-in properties for the external free store type. Depending on the accessibility of the content, users decide whether to run the plug-in on the server or client.

3.70.2 Properties

The external free store object type inherits all its properties from its supertype.

3.71 External store

Represents an external storage area accessible to the server but whose physical contents are stored outside the server.

3.71.1 Description

- Supertype: Store
- Subtypes: External File Store, External URL Store, External Free Store
- Internal name: dm_extern_store
- Object type tag: 60

An external store object represents an external storage area accessible to the server but the physical contents is stored outside the server. An external store object eliminates the need to transfer the content from client to server during a save operation. All subtypes of dm_extern_store operate in token mode.

3.71.2 Properties

The table describes the object properties.

Table 3-71: External store type properties

Property	Datatype	Single or repeating	Description
a_content_static	Boolean	S	<p>Controls how getFile methods behave when fetching content from an external storage area. For more information about the getFile method, see <i>OpenText Documentum Content Management - Administrator User Guide</i> (EDCAC250400-UGD).</p> <p>The default value is F.</p>

Property	Datatype	Single or repeating	Description
a_exec_mode	Boolean	S	<p>Controls where to execute the plug-in. Valid values are:</p> <ul style="list-style-type: none"> • <i>FALSE</i>: Directs the Foundation Java API to execute the plug-in on the server host. This is the default value and must be used for Foundation Java API version 6 or later. • <i>TRUE</i>: Deprecated value for this property. It executes the plug-in on the client host. Foundation Java API version 6 or later does not support executing the plug-in on the client host.
a_plugin_id	ID	R	The ID of the document containing the plug-in DLL (Windows) or shared library (Linux). The storage type for this document must be dm_filestore.
a_platform	integer	R	Specifies the client platform. Valid values are: <ul style="list-style-type: none"> • 1: Windows • 5: Macintosh • 6: Linux
a_storage_param_name	string(64)	R	A user-defined name of the attribute containing specific information about the store.

Property	Datatype	Single or repeating	Description
a_storage_param_value	string(1024)	R	The user-defined description of the a_storage_param_name attribute.
is_writable	Boolean	S	Specifies whether this external store supports writable operations.
storage_class	string(64)	S	Specifies for what the store is used.

3.72 External URL store

Represents an external storage area whose content is accessed using a token that follows the URL standard.

3.72.1 Description

- Supertype: External Store
- Subtypes: None
- Internal name: dm_extern_url
- Object type tag: 62

An external URL store object represents objects whose content is stored externally and accessed using token-mode operation, in which the token follows the URL standard. For the external URL store object type, it is best to configure the plug-in to retrieve content through the client. URLs are universal and can be accessed from anywhere. Documentum CM Server does not validate the URL.

3.72.2 Properties

The external URL store object type inherits all its properties from its supertype.

3.73 Federation

Describes a federation.

3.73.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_federation
- Object type tag: 5e

A federation object describes a federation. Federations are created using Documentum Administrator. Each repository belonging to a federation has a federation object that describes the federation.

3.73.2 Properties

The table describes the object properties.

Table 3-72: Federation type properties

Property	Datatype	Single or repeating	Description
r_govern_docbase	string(120)	S	<p>Name of the governing repository in the federation.</p> <p>This value serves as the default name of the federation unless the object's object_name property is explicitly set.</p> <p>This property is set in the federation objects for all members of the federation.</p>
r_govern_is_active	Boolean	S	<p>Indicates whether the governing repository is active. If this is FALSE (the default), federation-related jobs cannot be executed.</p> <p>This value is set only in the federation object in the governing repository.</p>

Property	Datatype	Single or repeating	Description
r_govern_suspended	Boolean	S	<p>Indicates whether transactions or changes to configuration objects can occur in the governing repository. The default is FALSE, meaning that transactions are allowed. This is used to manage synchronization and integrity between repositories in the federation while jobs are executing.</p> <p>This value is set only in the federation object in the governing repository.</p>
r_member_docbases	string(120)	R	<p>Names of the repositories that belong to the federation. Each name must be unique within the list.</p> <p>This value is set only in the federation object in the governing repository.</p>

Property	Datatype	Single or repeating	Description
r_member_is_active	Boolean	R	<p>Indicates whether the member repository is active in the management of configuration objects. The value at each index level is associated with the repository at the corresponding index level in r_member_docbases.</p> <p>The default is TRUE, meaning that the repository participates.</p> <p>This value is set only in the federation object in the governing repository.</p>
r_member_refresh	date	R	<p>Indicates the last refresh date of the configuration objects in the member repositories. The date at each index level is associated with the repository at the corresponding index level in r_member_docbases.</p> <p>This value is set only in the federation object in the governing repository.</p>

Property	Datatype	Single or repeating	Description
user_subtypes	string(32)	R	dm_user subtypes to propagate. The federation update job propagates users represented by the users subtypes listed in this property (in addition to users represented by dm_user objects). The subtypes must exist on all member repositories.

3.74 File store

Contains information about a file store storage area.

3.74.1 Description

- Supertype: Store
- Subtypes: None
- Internal name: dm_filestore
- Object type tag: 28

A file store object contains information about a file store storage area. A dm_filestore type storage area is one of the most common types of file storage areas in a Documentum CM Server installation. It is used to store files that have a wide variety of formats.

3.74.2 Properties

The table describes the object properties.

Table 3-73: File store type properties

Property	Datatype	Single or repeating	Description
is_public	Boolean	S	Specifies whether the area is accessible to the public without restrictions. Valid values are: <ul style="list-style-type: none">• 1: Accessible• 0: Not accessible

Property	Datatype	Single or repeating	Description
root	string(64)	S	The object name of the location object representing this storage area. Refer to the caution documented after this table.
use_extensions	Boolean	S	Specifies whether the server should append a DOS extension to the file when writing it into the storage area.



Caution

Be sure that the storage path you specify does not point to the same physical location as any other file stores. If two file stores use the same physical location, data loss may result.

3.75 Folder

Folder objects are used to organize repository content.

3.75.1 Description

- Supertype: SysObject
- Subtypes: Cabinet, Category, Room, Module, Topic, DSM Application, DSM Submission, XML Application
- Internal name: dm_folder
- Object type tag: 0b

The folder object organizes the contents a repository. All SysObjects and SysObject subtypes (except cabinets) must be stored either directly in a cabinet or in a folder. Folders, in turn, are stored in cabinets or other folders.

3.75.2 Properties

The table describes the object properties.

Table 3-74: Folder type properties

Property	Datatype	Single or repeating	Description
i_ancestor_id	ID	R	The object ID of the folders or cabinets that contain this folder directly or indirectly.
r_folder_path	With Oracle: string(740) With SQL Server: see Description With PostgreSQL: string(450)	R	<p>The folder paths for all locations to which the folder is linked.</p> <p>The length of the folder path property varies for SQL Server customers:</p> <ul style="list-style-type: none">• 450: For a new SQL Server repository.• 765: For an upgraded SQL Server repository not using Unicode. <p> Note: If you want to migrate the database from Oracle to SQL Server or PostgreSQL, use the ALTER TYPE DDL to increase the length to 740 characters.</p>

3.76 Foreign key

Describes the set of properties that define a foreign key.

3.76.1 Description

- Supertype: Relation Type
- Subtypes: None
- Internal name: dm_foreign_key
- Object type tag: 65

A foreign key object describes the set of properties that define a foreign key. Documentum CM Server creates foreign key objects when users define foreign keys for a type. Users cannot create foreign key objects.

3.76.2 Properties

The table describes the object properties.

Table 3-75: Foreign key type properties

Property	Datatype	Single or repeating	Description
child_attributes	string(32)	R	The child key name properties for the foreign key.
parent_attributes	string(32)	R	The parent key name properties for the foreign key.

3.77 Format

Records file format information.

3.77.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dm_format
- Object type tag: 27

A format object contains information about a file format recognized by Documentum CM Server. A predefined set of file formats is installed by default when a repository is configured.

3.77.2 Properties

The table describes the object properties.

Table 3-76: Format type properties

Property	Datatype	Single or repeating	Description
a_page_plugin_name	string(255)	S	<p>The name of the plug-in that implements the ability to retrieve designated content pages in this format.</p> <p>In this context, <i>pages</i> refers to the numbered pages of the content, not the position of the content within the files associated with an object.</p>
asset_class	string(32)	S	Specifies the kind of asset represented by this format, such as audio, video, and so on. This property is used by applications.
can_index	Boolean	S	Specifies whether this format can be full-text indexed.
com_class_id	string(38)	S	The class ID (CLSID) recognized by the Windows registry for a content type.
default_storage	ID	S	Specifies the default storage area for this type of content.
description	string(64)	S	User-defined description of the format.
dos_extension	string(10)	S	The DOS extension to use when copying a file in the format into the common area, client local area, or storage.

Property	Datatype	Single or repeating	Description
filename_modifier	string(16)	S	The modifier to append to create a unique file name.
format_class	string(32)	R	<p>Documentum CM Server sets this to one of the following values:</p> <ul style="list-style-type: none"> • <i>ft_always</i>: The content files in this format are always indexed. • <i>ft_preferred</i>: If there are multiple renditions of the content, this format is the preferred format for indexing. <p>This value can also be a user-defined value to identify the format class. For example, the xml, xsd, and xsl formats are XML and MSOffice classes.</p>
icon_index	integer	S	The index that locates the icon for the format in the icon resource file.
is_hidden	Boolean	S	Specifies whether format is hidden. This property is used by some OpenText Documentum CM clients.
mac_creator	string(4)	S	Information used internally for managing Macintosh resource files.
mac_type	string(4)	S	Information used internally for managing Macintosh resource files.
mime_type	string(256)	S	The Multimedia Internet Mail Extension (MIME) for the content type.

Property	Datatype	Single or repeating	Description
name	string(64)	S	Name of the format, for example, doc or tiff. Do not set this property to longer than 32 characters because the format names are used in the a_content_type property of dm_sysobject. If set to longer than 32 characters, the format name will cause errors.
richmedia_enable	Boolean	S	Indicates whether thumbnails, proxies, and metadata are generated for content in this format. By default, thumbnails, proxies, and metadata is generated for content in jpeg, mpeg, pdf, gif, and avi format.
topic_filter	string(64)	S	Obsolete.
topic_format	ID	S	Obsolete.
topic_format_name	string(64)	S	Obsolete.
topic_transform	Boolean	S	Specifies whether to transform the format before indexing.
win31_app	string(12)	S	Specifies which application to launch when users select content associated with the format object. Used by OpenText Documentum CM clients.

3.78 FT engine config

Records configuration information for an index server.

3.78.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_ftengine_config
- Object type tag: 08

An FT engine config object stores the parameters for a particular index server. There is one FT engine config associated with each full-text index object in a repository. Permissions to modify objects of this type are controlled by dm_fulltext_admin_acl object.



Caution

Changing the values in the param_name or param_value properties manually may result in undefined behavior during indexing or querying. It is recommended that you do not change these properties unless told to do so by OpenText Global Technical Services.

One exception to the rule about modifying the values in param_name and param_value, is to modify the permissions applied to FTDQL and non-FTDQL searches. Change the value of param_value for the param_name value, security_mode. Use one of the following values:

- BROWSE: Displays all results for which the user has at least BROWSE permission. If the user has BROWSE permission, the summary is blank.
- READ: Displays all results for which the user has at least READ permission.
- SUMMARY_BASED (default): Displays all results if SUMMARY is not in the select list. Displays results for which the user has at least READ permission if SUMMARY is selected.

3.78.2 Properties

The table describes the property definitions.

Table 3-77: FT engine config type properties

Property	Datatype	Single or repeating	Description
dsearch_config_host	string(128)	S	Specifies the fully qualified host name or IP address of the xPlore host that the index agent connects to.
dsearch_config_port	integer	S	Specifies the HTTP or HTTPS port that the index agent connects to.
dsearch_qrserver_host	string(128)	S	Specifies the fully qualified host name or IP address of the xPlore host that the Documentum CM Server query plug-in connects to.
dsearch_qrserver_port	integer	S	Specifies the HTTP or HTTPS port of the xPlore host that the Documentum CM Server query plug-in connects to.
filter_config_id	ID	S	The object ID of dm_ftfilter_config object that contains the filter settings for the index server.
ft_collection_id	ID	R	Reserved for future use.
ftsearch_security_mode	integer	S	Security mode for the xPlore index server: <ul style="list-style-type: none"> • 1—security check is done on index server • 0—security check is done on Documentum CM Server

Property	Datatype	Single or repeating	Description
load_balancer_enabled	Boolean	S	Specifies whether the load balancer is enabled or not in the HA active-active mode. By default, the value is FALSE.
param_name	string(64)	R	The names of the parameters that are passed to the engine.
param_value	string(255)	R	The values of the parameters that are passed to the engine. The value at each index position is associated with the name at the same index position in param_name.

3.79 FT filter config

Records full-text filter configuration information for an index server.

3.79.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_ftfilter_config
- Object type tag: 08

An FT filter config object stores the parameters for a particular full-text filter for an index server. Do not modify this type directly, but use the full-text index agent user interface to set the filter attributes.

3.79.2 Properties

The table describes the property definitions.

Table 3-78: FT filter config type properties

Property	Datatype	Single or repeating	Description
cabinets_to_exclude	string(255)	R	Objects linked to this cabinet are not full-text indexed.
folders_to_exclude	string(740)	R	Objects linked to this folder, or its subfolders, are not full-text indexed.
types_to_exclude	string(27)	R	Object types that will not be full-text indexed. However, subtypes of the listed types are not excluded from full-text indexing.

3.80 FT index agent config

Stores information about an index agent.

3.80.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_ftindex_agent_config
- Object type tag: 08

An FT index agent config object stores information about one index agent.

3.80.2 Properties

The table describes the object properties.

Table 3-79: FT index agent config type properties

Property	Datatype	Single or repeating	Description
active_connectors	string(32)	S	<p>Specifies whether the index agent is running in normal or migration mode. Valid values are:</p> <ul style="list-style-type: none"> • <i>normal</i>: Normal mode. • <i>reindex</i>: Migration mode. <p>The default value is normal.</p>
checkpoint_interval	integer	S	Reserved for future use.
connectors_batch_size	integer	S	<p>The number of queue items picked up by the queue reader in each polling interval.</p> <p>The default is 1000.</p>
docbase_connector_wait_time	Integer	S	<p>The frequency, in seconds, at which the index agent polls for events.</p> <p>The default is 60.</p>
exporter_queue_threshold	integer	S	<p>Used to control additions to the internal in-memory queue of the index agent. If the number of objects in the queue is equal to or exceeds the value in this property, then no more objects can be added to the queue.</p> <p>The default value is 500.</p>
exporter_thread_count	integer	S	<p>Number of concurrent exporter threads the index agent can run.</p> <p>The default value is 3.</p>

Property	Datatype	Single or repeating	Description
force_inactive	integer	S	<p>Indicates whether the agent is operating normally, or is shut down and waiting for a manual start. Valid values are:</p> <ul style="list-style-type: none">• 0: The agent is operating normally.• 1: The server does not start the agent until this property is set to 0 manually. <p>The default value is 0.</p>
index_name	string(64)	S	Name of the full-text index object associated with this index agent.
indexer_queue_threshold	integer	S	<p>Used to control additions to the internal, in-memory queue of the index agent. If the number of objects in the queue is equal to or exceeds the value in this property, then no more objects can be added to the queue.</p> <p>The default value is 500.</p>

Property	Datatype	Single or repeating	Description
object_name	string(255)	S	This is an inherited property. For ft index agent config objects, the value must be unique among all ft index agent config objects in the repository. The default value for the index agent installed with Documentum CM Server is <i>hostname_indexagentname</i> .
queue_user	string(255)	S	The user name used by the index agent to look for queued items. The default is dm_fulltext_index_user.
r_fail_time	Date	S	Reserved for future use.
r_is_active	integer	S	Specifies whether the index agent is active or inactive. Valid values are <ul style="list-style-type: none"> • 0: The agent is inactive. • 1: The agent is active.
r_last_done_time	Date	S	Reserved for future use.
r_last_work_time	Date	S	Reserved for future use.
r_modify_date	Date	S	This is an inherited property. The value is the last time the associated index agent updated the ft index agent config object.
r_start_time	Date	S	Reserved for future use.

Property	Datatype	Single or repeating	Description
r_stop_time	Date	S	Reserved for future use.
runaway_item_timeout	integer	S	Specifies the time, in seconds, a queue item remains in the internal queue before it is abandoned. The default value is 600.
save_queue_items	integer	S	Specifies whether successfully processed queue items are saved or destroyed. Valid values are: <ul style="list-style-type: none">• 0: The successfully processed queue items are destroyed.• 1: The successfully processed queue items are marked <i>done</i> and saved. The default is 0.
shut_down_requested	integer	S	Reserved for future use.

3.81 FT high water mark

Records the full-text high water mark for an index server.

3.81.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_ftwatermark
- Object type tag: 08

An FT high water mark object stores information used during full-text migration (reindexing). This object is for internal use only. This information is displayed by the full-text application user interface.

3.81.2 Properties

The table describes the property definitions.

Table 3-80: FT high water mark type properties

Property	Datatype	Single or repeating	Description
acl_maxid	ID	S	The maximum ACL object ID when the high water mark object is created.
acl_watermark	ID	S	The last ACL object ID that the index agent has processed.
current_indexing_sys_type	string(27)	S	The type of the current object being indexed.
failed_count	integer	S	The number of objects for which indexing the metadata or the content has failed.
filtered_count	integer	S	Total number of objects filtered from full-text indexing for this reindexing session.
group_maxid	ID	S	The maximum group object ID when the high water mark object is created.
group_watermark	ID	S	The last group object ID that the index agent has processed.
processed_count	integer	S	The total number of objects processed by index agent for full-text indexing.
submitted_count	integer	S	The total number of objects submitted by the connector worker thread.
success_docs	integer	S	The number of documents successfully indexed.

Property	Datatype	Single or repeating	Description
sysobj_maxid	ID	S	The maximum system object ID when the high watermark object is created.
sysobject_watermark	ID	S	The last system object ID that the index agent has processed.
systypes_to_index	string(27)	R	The list of system types that will be reindexed.
total_docs	integer	S	The total number of system objects, ACL objects, and group objects when the high water object is created.
warning_count	integer	S	The number of objects for which metadata has been indexed, but not content.

3.82 dm_ftquery_subscription

Represents a subscribed query.

3.82.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_ftquery_subscription
- Object type tag: 08

A dm_ftquery_subscription object represents subscription-specific information but not the saved query itself, which is contained in a dm_smart_list object.

3.82.2 Properties

The table describes the object properties.

Table 3-81: dm_ftquery_subscription type properties

Property	Datatype	Single or repeating	Description
frequency	CHAR(32)	S	How often the subscription is to run. Valid value is represented in frequency query subscription job parameter.
last_exec_date	TIME	S	Last date and time that the subscription was executed.
subscriber_name	CHAR(32)	S	Name of the user who is uses this subscription.

Property	Datatype	Single or repeating	Description
result_strategy	INTEGER	S	<p>Integer that indicates whether existing results that are saved in the dm_smart_list are to be replaced with the new results (0, the default), merged with the new results (1), or the new results are to be discarded (2).</p> <p>The saved query results are sorted as follows:</p> <ul style="list-style-type: none"> • If the result_strategy is set to 0, then the new results are sorted from the highest to lowest score. • If the result_strategy is set to 1, then the new results are listed first and sorted from the highest to lowest score; then the existing results are listed next and sorted from the highest to lowest score.
workflow_id	ID	S	Process ID of the workflow to be executed by the job. If this value is null, then the notification is executed through queue item creation when any result is returned.

Property	Datatype	Single or repeating	Description
zone_value	INTEGER	S	Zone to which this subscription belongs. With this value, all subscriptions with the same frequency can be picked by different jobs (User can customize jobs such that those jobs will be run on the same interval but with different value in job argument of "zone_value". Specify this value to run jobs when there are too many subscriptions for a single job.
oldest_queueitem_date	TIME	S	Records the oldest queue item date_sent value when subscription is run by QBS job. When job runs the same subscription next time, it will pick the value in oldest_queueitem_date if it is not null to cover index latency issue.

3.83 Fulltext index

Represents a full-text index for the repository.

3.83.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dm_fulltext_index
- Object type tag: 3b

A full-text index object represents the index associated with a repository. The object is created internally when a normal mode index agent is configured for the repository. Only the ft_engine_id property can be modified by a user. Superuser or sysadmin user privileges are required to change this property.

3.83.2 Properties

The table describes the object properties.

Table 3-82: Fulltext index type properties

Property	Datatype	Single or repeating	Description
codepage	string(64)	S	Specifies the code page that is used for strings sent to and received from the Index Agent. Typically, it is a UTF-8 code page.
ft_engine_id	ID	S	The object ID of the dm_ftengine_config object used for this index.
index_name	string(64)	S	The name of the full-text index object. The name must consist of ASCII characters.
install_loc	string(32)	S	The name of the location object that identifies the location of the fulltext installation for this index. The default name is the value of the fulltext_location property in the server config object.
is_standby	Boolean	S	Reserved for internal use.
r_content_count	integer	S	Reserved for future use.
r_index_type	integer	S	Specifies the type of index. The default value is 4.
r_last_clean	integer	S	Reserved for future use.
r_last_update	DATE	S	Time and date of the last update for this index.

3.84 Func expr

Stores the data dictionary information needed to execute expressions written in a third-party expression language.

3.84.1 Description

- Supertype: Expression
- Subtypes: Cond Expr
- Internal name: dm_func_expr
- Object type tag: 55

A func expr object stores the data dictionary information needed to execute expressions written in a third-party expression language. Documentum CM Server creates and manages func expr objects. Users cannot create func expr objects directly.

3.84.2 Properties

The table describes the object properties.

Table 3-83: Func expr type properties

Property	Datatype	Single or repeating	Description
attr_object_index	integer	R	<p>Specifies an index for the type_name list that identifies the property source type for each attribute_name property.</p> <p>The attr_object_index[0] value is the index value for the attribute_name[0] property. For example, if the attr_object_index[0] value is 3, the attribute_name[0] value is specified in the type_name[3] property.</p>

Property	Datatype	Single or repeating	Description
attribute_name	string(32)	R	List of property names that are passed to the expression. The names are passed in the order in which they are listed.
code_page	integer	R	Content page number of the object specified in the routine_id property. Currently, this value is always 1.
expression_kind	integer	S	Specifies whether the expression is a complete routine or an expression. Valid values are: <ul style="list-style-type: none">• 1: Expression• 2: User routine
expression_lang	integer	S	Specifies the language of the expression or routine. The only valid value is <ul style="list-style-type: none">• 1: Docbasic
object_alias	string(128)	R	Specifies an alias for each name listed in the type_name property. The aliases are used to map properties in the expression_text to their associated types. The alias at each index level is associated with the type specified at the corresponding index level in type_name.

Property	Datatype	Single or repeating	Description
repeat_attr_index	integer	R	<p>Provides an index for each repeating attribute_name value. Valid values are:</p> <ul style="list-style-type: none"> • <i>0 or greater</i>: Index into the repeating property. • -2: Any property value. • -3: All property values. • -4: First property value. • -5: Last property value. <p>For single-valued attribute_name properties, the corresponding repeat_attr_index property value is always -1.</p>
routine_id	ID	S	The object ID of the SysObject containing the source and pcode content pages.
routine_name	string(255)	S	The name of the entry point in the routine.
source_page	integer	S	<p>The content page number of the object specified in routine_id containing the source code.</p> <p>Currently, this value is always 0.</p>
type_name	string(32)	R	Object types names that are the sources of the properties specified in the attribute_name property.

3.85 Group

Contains information about a group in the repository.

3.85.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dm_group
- Object type tag: 12

A group object contains information about a group in the repository.

3.85.2 Properties

The table describes the object properties.

Table 3-84: Group type properties

Property	Datatype	Single or repeating	Description
alias_set_id	ID	S	The object ID of an alias set associated with the group.
Description	string(255)	S	A user-defined description of the group.
globally_managed	Boolean	S	Specifies whether the group is managed globally or locally. Valid values are: <ul style="list-style-type: none">• <i>TRUE</i>: The group is managed globally.• <i>FALSE</i>: The group is managed locally. The default value is <i>FALSE</i> . Changing the value requires at least Sysadmin privileges.
group_address	string(80)	S	The electronic mail address for the group.

Property	Datatype	Single or repeating	Description
group_admin	string(255)	S	The name of a user or group who can modify the group.
group_class	string(32)	S	Specifies the group type. Valid values are: <ul style="list-style-type: none"> • <i>group</i> • <i>role</i> • <i>module role</i> • <i>privilege group</i> • <i>domain</i>
group_directory_id	ID	S	The object ID of the LDAP config object representing the LDAP directory used to synchronize this group. This property is deprecated.
group_display_name	string(255)	S	The name of the group as it appears in an application. The name must be unique within the application. The default is the group_name.
group_global_unique_id	string(400)	S	Reserved for future use.

Property	Datatype	Single or repeating	Description
group_name	string(255)	S	<p>The name of the group.</p> <p>If the group_class property value is set to role, group name is the role name.</p> <p>If the group_class property value is set to domain, the group name is the domain name value.</p> <p>The name must consist of characters compatible with the server os code page of the Documentum CM Server and must be unique among the user and group names in the repository.</p> <p> Note: The special character ^ is used for delimitation of the group names. If you use the ^ or # symbols, the group function will not work as it causes a conflict with the delimitation symbol.</p>
group_native_room_id	ID	S	The object ID of the room that contains a private group.

Property	Datatype	Single or repeating	Description
group_source	string(16)	S	<p>Specifies the source of the group. The only valid value is LDAP, meaning the group was created by importing an LDAP group.</p> <p>This property is deprecated.</p>
groups_names	string(255)	R	The names of groups that are users in this group.
i_all_users_names	string(32)	R	<p>List of all users in the group, including those users that belong to groups contained within the group.</p> <p>This property value is computed when the property is first queried in a session and cached on the server and client.</p>
i_nondyn_supergroups_names	string(255)	R	The names of the non-dynamic groups to which this group belongs.
i_supergroups_names	string(255)	R	The name of the group and all groups that contain this group.
is_dynamic	Boolean	S	<p>Specifies whether the group is dynamic. Value values are:</p> <ul style="list-style-type: none"> • <i>T</i>: The group is a dynamic group. • <i>F</i>: The group is not a dynamic group. <p>The default value is FALSE.</p>

Property	Datatype	Single or repeating	Description
is_dynamic_default	Boolean	S	<p>Controls whether users in the of potential group users are considered members of the group by default when they connect to the repository. Valid values are:</p> <ul style="list-style-type: none">• <i>T</i>: Users are treated as members of the group.• <i>F</i>: Users are not treated as group members when they connect. <p>The default value is <i>F</i>.</p>
is_module_only	Boolean	S	<p>Specifies whether it is a dynamic group to which only privileged BOF modules can add members. Valid values are:</p> <ul style="list-style-type: none">• <i>T</i>: Only privileged BOF modules can add members to the group.• <i>F</i>: This is not a group to which only privileged BOF modules can add members. <p>The default value is <i>F</i>.</p>

Property	Datatype	Single or repeating	Description
is_protected	Boolean	S	<p>Specifies whether this is a dynamic group to which only authorized client installations can add members. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i>: Only authorized client installations can add members to the group. • <i>F</i>: This is not a group to which only authorized client installations can add members. <p>Client installation authorizations are recorded in the client rights database.</p>
is_private	Boolean	S	<p>Specifies whether the group is private or public. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i>: The group is private. • <i>F</i>: The group is public.
owner_name	string(255)	S	The name of the user or group who owns the group.
r_has_events	Boolean	S	Specifies whether someone has registered the group for auditing.
r_object_id	ID	S	The object ID of the group.
users_names	string(255)	R	The names of the users directly contained in the group.

3.86 Index

Stores information about an RDBMS index created in the repository.

3.86.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_index
- Object type tag: 1f

An index object stores information about an RDBMS index created in the repository.

3.86.2 Properties

The table describes the object properties.

Table 3-85: Index type properties

Property	Datatype	Single or repeating	Description
attribute	integer	R	The position of the properties being indexed in the type definition.
attr_count	integer	S	The number of properties (columns) in the index.
data_space	string(64)	S	The name of the Oracle tablespace or MS SQL Server database in which the index resides.
index_type	ID	S	The object ID of the dm_type object that represents the type that is indexed.
is_unique	Boolean	S	Specifies whether the index rows are unique. Value values are: <ul style="list-style-type: none">• <i>T</i>: The index rows are unique.• <i>F</i>: The index rows are not unique.

Property	Datatype	Single or repeating	Description
name	string(64)	S	The name of the index. The index name is generated internally.
rebuilding	Boolean	S	Specifies whether the index is rebuild. Valid values are: <ul style="list-style-type: none"> • <i>T</i>: The index is in the process of being rebuild. For example, if the index is currently moved to a different location. • <i>F</i>: The index is not being rebuild.
repeating	Boolean	S	Specifies whether the index is built on the single-valued type property table or the repeating type property table. Valid values are: <ul style="list-style-type: none"> • <i>T</i>: The index is built on the repeating type property table. • <i>F</i>: The index is built on the single-value property type table.
use_id_col	Boolean	S	Used by Documentum CM Server for server-generated indexes.
use_pos_col	Boolean	S	Used by Documentum CM Server for server-generated indexes.
use_tag	integer	R	Used by Documentum CM Server for server-generated indexes.

3.87 Jar

Represents a JAR file stored in the repository.

3.87.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dmc_jar
- Object type tag: 09

A jar object contains information about a JAR file stored in the repository as the object's content. The object type is installed by a script when Documentum CM Server is installed.

3.87.2 Properties

The table describes the object properties.

Table 3-86: Jar type properties

Property	Datatype	Single or repeating	Description
jar_type	integer	S	Specifies what type of jar file this object represents. Valid values are: <ul style="list-style-type: none">• 1: Interface• 2: Implementation• 3: Interface and implementation
minimum_vm_version	string(32)	S	Specifies the VM version level required by this file.

3.88 Java library

Represents a third-party Java library stored in the repository.

3.88.1 Description

- Supertype: Folder
- Subtypes: None
- Internal name: dmc_java_library
- Object type tag: 0b

Java library objects represent Java libraries stored in the repository. Each object represents one library. The object type is installed by a script when Documentum CM Server is installed.

3.88.2 Properties

The table describes the object properties.

Table 3-87: Java library type properties

Property	Datatype	Single or repeating	Description
sandbox_library	Boolean	S	<p>Specifies how the library is loaded. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i>: The library is loaded by the Business Object class loader. • <i>F</i>: The library loaded by the Shared class loader.

3.89 JMS config

Specifies the properties required for Java Method Server.

3.89.1 Description

- Supertype: Sysprocess Config
- Subtypes: None
- Internal name: dm_jms_config
- Object type tag: 08

All properties of dm_sysobject also belong to dm_jms_config.

This table describes the additional properties.

Table 3-88: General JMS config type properties

Property	Datatype	Single or repeating	Description
base_uri	string(255)	R	<p>URI for the servlet at the corresponding index position in servlet_name. Contains the host, port, and servlet_path in the following format: http://host:port/servlet_path</p> <p>host is the IP address or name of the machine hosting the application server.</p> <p>port is the port number on which the application server is listening.</p> <p> servlet_path is the path to the servlet to which an HTTP_POST request is directed.</p> <p>The URI can contain only ASCII characters.</p>
config_type	integer	S	<p>Specifies whether the accessible storage areas and defined projection targets are configured in the server config object or this JMS config object. Valid values are:</p> <ul style="list-style-type: none"> • 1: Server config object erver config object (identified in svr_config_id). • 2: JMS config object (defined in the near_stores and projection-related properties).

Property	Datatype	Single or repeating	Description
is_disabled	Boolean	R	Specifies whether the topic is disabled. T means the topic is disabled. F means the topic is not disabled. The default is F.
network_locations	string(80)	R	The network identifiers of the network locations served by the JMS.
projection_enable	Boolean	R	Indicates whether projection to the connection broker specified at the corresponding index position in projection_targets is enabled.
projection_netloc_enable	Boolean	R	Indicates whether projection to the network location specified at the corresponding index position in projection_netloc_id is enabled.
projection_netloc_ident	string(80)	R	User-defined identifiers of network locations.
projection_ports	integer	R	Specifies the port on which the connection broker is listening. The value at each index level is matched to the connection broker specified at the corresponding level in projection_targets. The default is 1489.

Property	Datatype	Single or repeating	Description
projection_proximity_value	integer	R	<p>Proximity values projected to the connection brokers, the network locations, or both. The value at each index position is projected to the connection broker specified at the corresponding position in projection_targets if projection_enable is TRUE in the corresponding index position.</p> <p>Similarly, the value at each index position is projected to the network location specified at the corresponding index position in projection_netloc_id if projection_netloc_enable is TRUE in the corresponding index position.</p>
projection_targets	string(80)	R	Names of the host machines on which the connection brokers reside.
server_config_id	ID	R	Object id of server config objects from which current JMS can be used.
server_major_version	string(16)	R	Used internally.
server_minor_version	string(16)	R	Used internally.
servlet_name	string(32)	R	Name of a Java servlet. The name do_method identifies the servlet to which DO_METHOD functions are directed.

Property	Datatype	Single or repeating	Description
supported_protocol	string(6)	R	The network protocols supported by the JMS server. Values in this property must all be lowercase. For example, http or https.

3.90 Job

Used to run a program at regularly scheduled intervals.

3.90.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_job
- Object type tag: 08

A job object represents a program that runs at regularly scheduled intervals. A job object stores information about the program, such as the name of the method object for the program, how often or when to run it, the number of times the program has been run, the date of its last execution, and so on. Programs represented by job objects are launched by a special utility called the dm_agent_exec utility, which is itself started by Documentum CM Server.

3.90.2 Properties

The table describes the object properties.

Table 3-89: Job type properties

Property	Datatype	Single or repeating	Description
a_continuation_interval	integer	S	Specifies how often to re-invoke the job, in minutes. If set to a number greater than zero, the job is automatically restarted after the specified interval. The default value is 0.

Property	Datatype	Single or repeating	Description
a_current_status	string(255)	S	Contains job status information.
a_is_continued	Boolean	S	Specifies whether the current job was invoked automatically. Valid values are: <ul style="list-style-type: none"> • <i>T</i>: The current job was invoked automatically. • <i>F</i>: The current job was not invoked automatically. The default is <i>F</i> .
a_iterations	integer	S	The number of times the job has been invoked and completed.
a_last_completion	date	S	Contains the date and time when the last invocation of the job was completed.
a_last_document_id	ID	S	The object ID of the status document created for the last invocation of the job.
a_last_invocation	date	S	Contains the date and time when the last invocation of the job was started.
a_last_process_id	ID	S	The process ID of the most recent dm_method object.
a_last_return_code	integer	S	The return status of the most recent dm_method object.
a_next_continuation	date	S	The date of the next automatic job invocation. The default date is NULLDATE.
a_next_invocation	date	S	The computed time of the next invocation.

Property	Datatype	Single or repeating	Description
a_special_app	string(32)	S	This property is inherited from the dm_sysobject. It is set by the server when the job is checked out by the dm_agent_exec utility for execution.
expiration_date	date	S	The date and time on which the job expires. The job is not be executed after the specified date and time. The default date is NULLDATE.
inactivate_after_failure	Boolean	S	Specifies whether the job is inactive due to failure. valid values are: <ul style="list-style-type: none"> • <i>T</i>: The job is inactive because it failed. • <i>F</i>: The job is active. The default is F.
is_inactive	Boolean	S	Specifies whether the job is unavailable for execution. Valid values are: <ul style="list-style-type: none"> • <i>T</i>: The job is inactive and cannot be executed. • <i>F</i>: The job is active and can be executed. The default is F.
max_iterations	integer	S	The maximum number of times to execute the job. The default is 1. A value of zero (0) indicates that there is no maximum.

Property	Datatype	Single or repeating	Description
method_arguments	string(255)	R	The arguments to pass to the method. These arguments are passed only if the pass_standard_arguments property is set to F.
method_data	string(255)	R	Used by the procedure associated with the job. The procedure can modify this property.
method_name	string(255)	S	The name of the method to invoke when the job is executed. This is a required property.
method_trace_level	integer	S	Controls tracing for the method. Valid values are: <ul style="list-style-type: none"> • 0: Disable tracing. • Any integer value greater than 0: Enable tracing.
object_name	string(255)	S	The name of the job object. This is inherited from the dm_sysobject.

Property	Datatype	Single or repeating	Description
pass_standard_arguments	Boolean	S	<p>Specified whether to pass standard arguments. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i>: The standard arguments are passed to the method and the arguments in method_arguments are not passed. • <i>F</i>: The standard arguments are not passed to the method, but the arguments in the method_arguments value are passed instead. <p>The standard arguments are: - docbase_name repositoryname.serverconfig - user_name username job_id jobid - method_trace_level methodtracelevel</p> <p>The default is F.</p>
run_interval	integer	S	Used in conjunction with the run_mode property to determine how often to invoke the job. The default value is 0.

Property	Datatype	Single or repeating	Description
run_mode	integer	S	<p>Used in conjunction with the run_interval property to determine how often to invoke the job. Valid values are:</p> <ul style="list-style-type: none"> • 0: Never. • 1: Every x minutes. • 2: Every x hours. • 3: Every x days. • 4: Every x weeks. • 5: Every x months. • 7: Every x day of week. • 8: Every x day of month. • 9: Every x day of year. <p>Where x is the value specified in the run_interval property.</p> <p>The default value is 0.</p>
run_now	Boolean	S	<p>Specifies whether to run the job the next time the agent exec program polls the repository. Value values are:</p> <ul style="list-style-type: none"> • T: The job runs the next time the repository is polled. • F: The job is not run the next time the repository is polled.
start_date	date	S	The date and time of the first required invocation. This is a required property.

Property	Datatype	Single or repeating	Description
target_server	string(255)	S	<p>Specifies a particular server to run the job. The server name must be specified in the following format: repository[.server_config_name][@machine]</p> <p>The default value is NULLSTRING.</p>

3.91 Job request

Used internally by Documentum Administrator.

3.91.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_job_request
- Object type tag: 08

A job request object records information needed by Documentum Administrator to execute certain jobs. Users cannot create these objects manually.



Caution

Users, or applications, must not modify these objects. However, these objects can be deleted if the request_completed property is set to T, indicating that the associated job has been invoked.

3.91.2 Properties

The table describes the object properties.

Table 3-90: Job request type properties

Property	Datatype	Single or repeating	Description
arguments_keys	string(255)	R	Internal keys used in conjunction with the arguments_values property. The key at any particular index position is used in conjunction with the value at the corresponding index position in the arguments_values property.
arguments_values	string(255)	R	The argument values passed to the job.
job_name	string(255)	S	The name of the requested job.
method_name	string(255)	S	The name of the method invoked by the job.
priority	integer	S	A priority value internally assigned by Documentum Administrator.
request_completed	Boolean	S	Specifies whether the job has been executed. Valid values are: <ul style="list-style-type: none">• <i>T</i>: The job has been invoked.• <i>F</i>: The job is awaiting execution.

3.92 Job sequence

Specifies a job that belongs to a set of jobs executed in a particular sequence.

3.92.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_job_sequence
- Object type tag: 08

A job sequence object stores information about a job that belongs to a set of jobs executed in a particular sequence. Sequenced jobs are executed using the `dm_run_dependent_jobs` method, which can be invoked by another, controlling job or on the command line. Job sequence objects are created using Documentum Administrator.

3.92.2 Properties

The table describes the object properties.

Table 3-91: Job sequence type properties

Property	Datatype	Single or repeating	Description
job_display_name	string(255)	S	Reserved for use by Documentum Administrator.
job_docbase_name	string(255)	S	<p>Specifies the repository in which the job resides. This string is used to match an entry in the repository connection file if that file is used.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • <code>repository_name</code> • <code>repository_name.content_server_name</code> • <code>repository_name@host_name</code> • <code>repository.content_server_name@host_name</code>
job_id	ID	S	The object ID of the job.

Property	Datatype	Single or repeating	Description
job_login_domain	string(255)	S	The login domain of the user identified in the job_login_user_name property.
job_login_user_name	string(255)	S	The login name of the job.
object_name	string(255)	S	This inherited property must be set to the name of the job sequence that includes the job identified in this job sequence object.
predecessor_id	ID	R	<p>Specifies the job or jobs that must complete successfully before the job specified in job_id is run.</p> <p>The jobs are identified by their object IDs.</p> <p>This property can be empty.</p>

3.93 Key

Contains information describing a key for an object type.

3.93.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dm_key
- Object type tag: 59

A key object contains information describing a key for an object type. Documentum CM Server creates key objects when users define keys for a type. Users cannot create key objects.

3.93.2 Properties

The table lists the properties defined for the type.

Table 3-92: Key type properties

Property	Datatype	Single or repeating	Description
is_unique	Boolean	S	Indicates whether the key is a unique key.
key_attributes	string(32)	R	Names of the properties in the key definition.
type_name	string(32)	S	Name of the object type containing the key properties.
parent_id	ID	S	Object ID of an aggrdomain object. Used internally by dump and load operations.

3.94 LDAP config

Stores the set-up values that configure an LDAP-compliant directory server.

3.94.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_ldap_config
- Object type tag: 08

An ldap config object stores the set-up values that configure an LDAP-compliant directory server. The set-up values are defined through Documentum Administrator when the repository is set up to use the LDAP directory server. Setting the values in an ldap config object requires superuser user privileges.

3.94.2 Properties

The table describes the object properties.

Table 3-93: LDAP config type properties

Property	Datatype	Single or repeating	Description
a_last_run	string(32)	S	Used internally.
a_last_no	string(20)	S	Used internally.
bind_dn	string(255)	S	A distinguished name for binding. This is a required property.
bind_pwd	string(32)	S	The password for the binding name. This is a required property.

Property	Datatype	Single or repeating	Description
bind_type	string(16)	S	<p>Defines the binding mode used to authenticate LDAP users. Valid values are:</p> <ul style="list-style-type: none"> • <i>bind_search_dn</i>: Uses the operating system name to find the Distinguished name for authentication. • <i>bind_by_dn</i>: Uses the Distinguished name stored in the <i>user_ldap_dn</i> property to perform authentication. <p> Notes</p> <ul style="list-style-type: none"> • By default, the <i>dm_LDAPSynchronization</i> job imports the <i>dm_user</i> attribute <i>user_ldap_dn</i> to the repository. If you do not want to import this attribute, set the <i>dm_ldap_config.bind_type</i> to <i>bind_search_dn</i> and set the job argument <i>import_user_ldap_dn</i> to <i>false</i>. <pre>API> retrieve,c, dm_ldap_config API> set,c,1,bind_type API> bind_search_dn API> save,c,1 API> retrieve,c,dm_jobwhere object_name='dm_LDAPSynchronization' API> append,c,1,method_arguments API> -import_user_ldap_dn false API> save,c,1</pre> If you do not specify this argument, the default value (<i>true</i>) is considered and the attribute gets imported to the repository. • During the synchronization of groups and members, the authentication through <i>bind_search_dn</i> could cause authentication failure for users who are not identified by the user search filter. To avoid authentication failure in such scenarios, use <i>bind_by_dn</i>.
certdb_location	string(255)	S	The name of the location object that points to the certificate database directory.

Property	Datatype	Single or repeating	Description
deactivate_user_option	Boolean	S	<p>Specifies whether the synchronization job deactivates users that have been deactivated in the Directory Server. Value values are:</p> <ul style="list-style-type: none"> • <i>T</i>: The synchronization job deactivates users that have been deactivated in the Directory Server. • <i>F</i>: Disables a repository update. <p>The default value is <i>F</i>.</p>
failover_ldap_config_ids	ID	R	The object IDs of the ldap config objects representing the LDAP directory servers used as secondary (failover) servers for this LDAP directory server.
failover_use_interval	time	S	<p>Specifies the length of time, in minutes, during which Documentum CM Server uses a secondary LDAP directory server for user authentication before attempting to contact the primary LDAP directory server.</p> <p>The default is 5 minutes.</p>
first_time_sync	Boolean	S	<p>Specifies whether the LDAP directory must perform a first-time synchronization of the users and groups in the repository that match users and groups in the directory. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i>: Synchronization • <i>F</i>: No synchronization <p>The default value is <i>F</i>.</p>
group_obj_class	string(64)	S	Specifies the object class of groups, as defined in the LDAP directory server. This property is optional. The default is <i>groupofuniquemembers</i> .
grp_search_base	string(256)	S	Specifies the starting point for group searches in the directory server schema. For more information about search bases, refer to the documentation provided by the directory server vendor.
grp_search_filter	string(256)	S	Specifies a filter for group searches. For more information about filter specifications, refer to the documentation provided by the directory server vendor.

Property	Datatype	Single or repeating	Description
group_tree_sync	Boolean	S	<p>Specifies whether to enable nested group synchronization or not.</p> <p>This property is set up by administrators using Documentum Administrator.</p> <p>Nested group synchronization is enabled only if the import_mode property is set to "group" or "all".</p>
import_mode	string(7)	S	<p>Controls whether LDAP synchronization is performed for users, groups, or both. Valid values are:</p> <ul style="list-style-type: none"> • <i>user</i> • <i>group</i> • <i>both</i> <p>The default value is <i>both</i>.</p>
ldap_host	string(128)	S	The host name of the machine on which the LDAP directory server is running. This is a required property.
map_attr	string(32)	R	<p>Specifies object properties that are set when the user or group is imported into the repository.</p> <p>The property at a particular index position is set to the value specified at the same index position in the map_val property.</p>
map_attr_type	string(32)	R	The object type for which the property identified at the corresponding index position in map_attr is defined.
map_const_attr	string(32)	R	Currently unused.
map_const_val	string(32)	R	Currently unused.

Property	Datatype	Single or repeating	Description
map_rejection	integer	R	<p>Specifies the rejection rule for a mapped property. If the value mapped to the property fails to satisfy the rule associated with the property, the user or group entry is rejected and is not synchronized in the repository. Valid values are:</p> <ul style="list-style-type: none"> • 0: Synchronization always occurs even if an LDAP attribute referenced in the mapping is missing, empty, or does not have sufficient characters. • 1: Synchronization does not occur if an LDAP attribute in the mapping is missing or empty. However, synchronization will occur even if an LDAP attribute in the mapping has fewer characters than specified in the mapping. • 2: Synchronization does not occur if an LDAP attribute referenced in the mapping is missing, empty, or has fewer characters than specified in the mapping. <p>For mandatory properties, both users and groups, the map rejection value is automatically set to 2 and cannot be changed.</p>
map_val	string(64)	R	The name of a Directory server property or a hard-coded value to which the property identified in the corresponding index position in map_attr is mapped.
map_val_type	string(1)	R	<p>Specifies whether the value in the corresponding index position in map_val is a Directory Server attribute name, an expression, or an actual value. Valid values are:</p> <ul style="list-style-type: none"> • A: The value is a Directory Server property. • E: The value is an expression. • V: The value is an actual value.
per_search_base	string(256)	S	Defines the starting point for user searches in the directory server schema. For more information about search bases, refer to the documentation provided by the directory server vendor.

Property	Datatype	Single or repeating	Description
per_search_filter	string(256)	S	Defines a filter for user searches. For more information about filter specifications, refer to the documentation provided by the directory server vendor.
person_obj_class	string(64)	S	Specifies the object class of users as defined in the LDAP directory server. This is an optional property. The default is <i>person</i> .
port_number	integer	S	The port number on the LDAP directory server host machine which Documentum CM Server uses to communicate with the directory server.
rename_group_option	Boolean	S	Specifies whether group names are updated in the repository during synchronization if they have changed in the Directory Server. Valid values are: <ul style="list-style-type: none"> • <i>T</i>: Group names are updated. • <i>F</i>: Group names are not updated. The default value is FALSE.
rename_user_option	Boolean	S	Specifies whether user names are updated in the repository during synchronization if they have changed in the Directory Server. Valid values are: <ul style="list-style-type: none"> • <i>T</i>: User names are updated. • <i>F</i>: User names are not updated. The default value is FALSE.
retry_count	integer	S	Specifies the number of times Documentum CM Server attempts to contact the LDAP directory server before reporting that it cannot connect. The default value is 3.
retry_interval	time	S	Specifies the time interval, in seconds, that must elapse before Documentum CM Server tries to contact the LDAP directory server again. The default value is 5 seconds.

Property	Datatype	Single or repeating	Description
ssl_mode	integer	S	<p>Specifies whether LDAP user authentication takes place over a secure socket layer (SSL). Valid values are:</p> <ul style="list-style-type: none"> • 0: SSL is not used. • 1: SSL with server authentication is used. <p>The default value is 0.</p>
ssl_port	integer	S	<p>The port number of the LDAP SSL port. The default is port is 636.</p>
use_ext_auth_prog	Boolean	S	<p>Specifies whether Documentum CM Server uses an external password checking program to authenticate LDAP users. Valid values are:</p> <ul style="list-style-type: none"> • T: An external password checking program is used. • F: Documentum CM Server authenticates LDAP users. <p>The default is FALSE.</p>
user_subtype	string(64)	S	<p>Specifies the object type of a user if the user is a subtype of the dm_user object.</p>

3.95 Link record - Deprecated

Contains information about established links to content storage areas.

3.95.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_linkrecord
- Object type tag: 2b

A link record object contains information about established links to content storage areas.



Caution

The dmi_linkrecord object type is deprecated. Foundation Java API version 6 does not support linked storage areas or manage link record objects.

3.95.2 Properties

The table describes the object properties.

Table 3-94: Link record type properties

Property	Datatype	Single or repeating	Description
parent_id	ID	S	The object ID of the associated content object.
component_id	ID	S	The object ID of the content object associated with the content.
format_id	ID	S	The object ID of the format object representing the format of the linked content.
data_ticket	integer	S	Used internally to fetch and save the content.
other_ticket	integer	S	Used internally to fetch and save the content.
session_count	integer	S	The number of sessions that are accessing the content storage area identified by this link record object.
session_id	ID	R	The session IDs of the sessions accessing the content storage area.

3.96 Linked store - Deprecated

A linked store object represents a storage area that links a common directory accessible to all clients and an actual file directory.

3.96.1 Description

- Supertype: Store
- Subtypes: None
- Internal name: dm_linkedstore
- Object type tag: 2a

A linked store object contains information about the link between a common directory accessible to all clients and an actual file directory.



Caution

The linked store object type is deprecated as of Foundation Java API version 6.0. Foundation Java API version 6 does not support linked store storage areas.

3.96.2 Properties

The table describes the object properties.

Table 3-95: Linked store type properties

Property	Datatype	Single or repeating	Description
link_location	string(64)	S	The name of the location object that represents the directory containing the logical link.
symbolic_links	Boolean	S	Specifies whether the links defined in the link directory are symbolic or hard links. Valid values are: <ul style="list-style-type: none">• <i>T</i>: Symbolic links.• <i>F</i>: Hard links.

3.97 Literal expr

Describes a literal value that is an integer, string, Boolean, ID, or date datatype.

3.97.1 Description

- Supertype: Expression
- Subtypes: None
- Internal name: dm_literal_expr
- Object type tag: 53

A literal expr object describes a literal value that is an integer, string, Boolean, ID, or date datatype. Literal expr objects are created and managed by Documentum CM Server and cannot be created by users.



Note: A date literal value specified in a literal expr object cannot require a pattern definition.

3.97.2 Properties

The literal expr type has no defined properties. All of its properties are inherited from the dm_expression supertype.

3.98 Load object record

Records information about objects that have been loaded from a dump file into a repository.

3.98.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_load_object_record
- Object type tag: 32

A load object record object contains information about one object that has been loaded from a dump file into a repository.

3.98.2 Properties

The table describes the properties definitions.

Table 3-96: Load object record type properties

Property	Datatype	Single or repeating	Description
extra_offset	string(20)	S	If the dump file is 2 GB or greater, this property specifies the position at which the server wrote the last entry.
extra_pos	integer	S	Specifies the position in the dump file, in number of characters, where that contains additional information about the object.
file_offset	string(20)	S	If the dump file is less than 2 GB, this property specifies the position at which the server wrote the last entry.
is_synonym	Boolean	S	Specifies whether the new_id property already exists in the new repository. This property is used for objects such as content storage areas and partitions, where the old object is not copied into the new repository, but merely used a matching object in the new repository. Valid values are: <ul style="list-style-type: none">• <i>T</i>: The new_id property already exists.• <i>F</i>: The new_id property does not exist.

Property	Datatype	Single or repeating	Description
load_object	ID	S	The object ID of the load record object that contains the information about the dump file containing the loaded object.
new_id	ID	S	The object ID of the loaded object in the new repository.
old_id	ID	S	The object ID of the loaded object in the old repository.

3.99 Load record

Stores information about dump files that are being loaded into a new repository.

3.99.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dm_load_record
- Object type tag: 31

A load record object stores information about a dump file that is being loaded into a new repository. The information in a load record object can be used in conjunction with the information in the associated load object record objects to restart the loading process if it is interrupted.

3.99.2 Properties

The table describes the object properties.

Table 3-97: Load record type properties

Property	Datatype	Single or repeating	Description
file_name	string(255)	S	The name of the dump file that is loaded.
load_operation	string(255)	S	Used internally.

Property	Datatype	Single or repeating	Description
load_parameter	string(255)	R	<p>Specifies the parameters for the load operation. Valid parameters and values are:</p> <ul style="list-style-type: none"> • preserve_replica= T F • generate_event=T F <p>The generate_event parameter is enabled (T) by default. Disabling the generate_event parameter (F) turns off the notification to the full-text index user for the save event when an object is loaded into the target repository.</p>
predicate	string(255)	R	Used internally.
r_offset	string(20)	S	Used internally to manage the load operation and objects created during the load.
relocate	Boolean	S	<p>Specifies whether the server assigns new object IDs to all the objects in the dump file during the load process. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i>: The server assigns new object IDs. • <i>F</i>: The server uses the previous object ID. <p>The default is <i>T</i>.</p>
r_end_time	date	S	The date and time at which the load process ended.

Property	Datatype	Single or repeating	Description
r_first_phase	Boolean	S	Specifies whether the load process has completed the first phase. Valid values are: <ul style="list-style-type: none"> • <i>T</i>: The first phase of the load process completed. • <i>F</i>: The first phase of the load process did not complete.
r_position	integer	S	Specifies how far into the dump file the load process has gone. This value is updated continually as the load process proceeds.
r_start_time	TIME	S	The time at which the load process was started.
type	string(33)	R	Used internally.

3.100 Location

Records a file system location for a specific file or directory.

3.100.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_location
- Object type tag: 3a

A location object contains a file system location for a specific file or directory. The server uses the information in location objects to find the files and directories that it needs for successful operation.

3.100.2 Properties

The table describes the object properties.

Table 3-98: Location type properties

Property	Datatype	Single or repeating	Description
file_system_path	string(255)	S	<p>Specifies the actual location of the directory or file represented by the location object. The syntax must be appropriate for the machine on which the server resides.</p> <p>For example, if the server is on a Windows host, the specification must be expressed as a Windows path specification. On a Linux host, the specification must be a Linux path.</p> <p>The directory path and name must consist of ASCII characters.</p> <p>Refer to the caution documented after this table.</p>
mount_point_name	string(32)	S	Specifies the mount point underneath which this location resides. Use the name of the mount point object that describes the mount point.
no_validation	Boolean	S	<p>Specifies whether applications create a location object. Valid values are:</p> <ul style="list-style-type: none">• <i>T</i>: The server does not validate the directory path specified by file_system_path and does not create the directory.• <i>F</i>: The server validates the path and creates a directory. <p>The default is FALSE.</p>
object_name	string(255)	S	<p>The name of the location object. The name must be unique among the location objects in a repository.</p> <p>This property is inherited from the SysObject supertype, but is included here because the setting has specific requirements for the location type.</p>

Property	Datatype	Single or repeating	Description
path_type	string(16)	S	Specifies whether the location represents a directory or a file. Valid values are: <ul style="list-style-type: none">• <i>directory</i>• <i>file</i>
security_type	string(32)	S	Specifies the security level applied to the directory or file. Valid values are: <ul style="list-style-type: none">• <i>public_open</i>• <i>public</i>• <i>private</i> The default setting is private.



Caution

Be sure that the combination of the mount point and path you specify does not point to the same physical location as any other file store. If two file stores use location objects that point to the same physical location, data loss may result.

3.101 Locator

Stores a Documentum Resource Locator (DRL), which points to an object in a repository.

3.101.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_locator
- Object type tag: 08

A locator object stores a DRL, which points to an object in a repository. The DRL is stored in the one property defined for the type.

3.101.2 Properties

The table describes the object properties.

Table 3-99: Locator type properties

Property	Datatype	Single or repeating	Description
resource_locator	string(255)	S	Contains a text string that resolves to an object in a repository.

3.102 Media profile

Stores a transformation profile as content.

3.102.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_media_profile
- Object type tag: 08

A media profile object stores a transformation profile as content. Transformation profiles are XML documents that describe the transformations that change the content format. A default profile set is provided when a Content Transformation Server product is configured for the repository. System administrators can create additional profiles.

3.102.2 Properties

The table describes the object properties.

Table 3-100: Media profile type properties

Property	Datatype	Single or repeating	Description
cmd_file_paths	string(255)	R	Internal property used by OpenText™ Documentum™ Content Management Transformation Services.

Property	Datatype	Single or repeating	Description
filter_names	string(32)	R	Internal property that specifies the filter Content Transformation Services uses. Valid values are: <ul style="list-style-type: none"> • <i>CTSProduct</i> • <i>Public</i>
filter_values	string(64)	R	Specifies whether the transformation is viewable in the WDK transformation wizard. Valid values are: <ul style="list-style-type: none"> • An acronym representing a OpenText Documentum Content Management (CM) Transformation Services product. • <i>System</i> • <i>Public</i>
related_objects_only	Boolean	S	Specifies whether a transformation can store the target file as a new object or a rendition. Valid values are: <ul style="list-style-type: none"> • <i>T</i>: The target only be stored as a new object. • <i>F</i>: The target can be stored as a new object or a rendition.
source_formats	string(64)	R	The names of the source formats identified in the transformation profile.
src_object_type	string(32)	R	The object types of the sources.

Property	Datatype	Single or repeating	Description
target_formats	string(64)	R	<p>The names of the target formats identified in the transformation profile.</p> <p>The format specified in any index position in the source_formats property can be transformed to the format identified in the corresponding index position in target_formats.</p>

3.103 Message address

Records all unique, full e-mail addresses for an e-mail message.

3.103.1 Description

Internal name: dm_message_address

This type was changed to a registered table in release 6.6, and its use is not supported for customer applications.

3.104 Message archive

Stores an e-mail message.

3.104.1 Description

- Type category: lightweight type
- Supertype: Message Container (shareable parent type)
- Subtypes: None
- Internal name: dm_message_archive
- Object type tag: 09

A message archive object is used to store an e-mail message. The header information in the message is stored in the object properties and the body of the message is stored as content. Message archive objects are primarily used by personal and compliance archiving applications. In release 6.5, this type was converted to a lightweight type that uses Message Container as its shareable parent type. Some of the type properties were moved to the parent type.

3.104.2 Properties

The table describes the object properties.

Table 3-101: Message archive type properties

Property	Datatype	Single or repeating	Description
attachment_count	integer	S	Total number of message attachments in the e-mail message.
conversation_id	string(32)	S	Reserved for internal use.
is_encrypted	string(1)	S	Specifies whether the message is encrypted. Valid values are: <ul style="list-style-type: none"> • 0: No encryption. • 1: MIME encryption. • 2: Other type of encryption.
message_id	string(42)	S	E-mail message identifier.
message_id_timestamp	string(8)	S	Reserved for internal use.
message_importance	string(1)	S	Specifies the importance level of the e-mail message. Valid values are: <ul style="list-style-type: none"> • 0: Normal • 1: Low • 3: Medium • 5: High
message_link_count	integer	S	The link number in a SysObject chain.
message_sensitivity	string(1)	S	Specifies the sensitivity of the message. Valid values are: <ul style="list-style-type: none"> • 0: None • 1: Personal • 2: Private • 3: Confidential

Property	Datatype	Single or repeating	Description
message_size	integer	S	Specifies the size of the complete e-mail message, including headers and routing information, in bytes.
message_subject	string(256)	S	The subject line of the message.
parent_message_id	string (24)	S	The object ID of the root message, if this message is an embedded message.
receive_date	Date	S	The GMT date and time at which the message was received.
sent_date	Date	S	The GMT date and time at which the message was sent.
tracking_id	string(2000)	S	Used internally to track the message source.

3.105 Message attachment

Records names of attachments on email messages.

3.105.1 Description

Internal name: dm_message_attachment

This type was changed to a registered table in release 6.6, and its use is not supported for customer applications.

3.106 Message queue config

A message queue config object contains the configuration properties of a message queue.

3.106.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_mq_config
- Object type tag: 08

Message queue config objects are created when an internal application creates a message queue. Its use is not supported for customer applications.

If an internal application creates a message queue, you may find objects of type dmc_mq_<Queue_Name>, where <Queue_Name> represents the name of the queue. These types and objects are also not supported for customer use.

3.107 Message container

Stores multiple e-mail messages.

3.107.1 Description

- Type category: shareable type
- Supertype: Document
- Subtypes: Message Archive (lightweight)
- Internal name: dm_message_container
- Object type tag: 09

A message container object is used as the shareable parent of message archive objects that store e-mail messages. Common information for groups of messages is stored in the object properties. The information for individual messages is stored in individual message archive objects, and the message body is stored as content.

Message container objects are primarily used by personal and compliance archiving applications.

3.107.2 Properties

The table describes the object properties.

Table 3-102: Message container type properties

Property	Datatype	Single or repeating	Description
datamodel_version	string(6)	S	Used internally to manage data model changes.

Property	Datatype	Single or repeating	Description
is_archive_message	string(1)	S	<p>Specifies whether the message is an archive message. Valid values are:</p> <ul style="list-style-type: none"> • 0: The message is not an archive message. • 1: The message is an archive message, and the message_id is deterministic. <p>Archive messages are immutable. Non-archive messages can be edited. The message_id is set internally and is unique to this message.</p>
message_class	string(1)	S	<p>Specifies the class of message represented by the mail message object. Valid values are:</p> <ul style="list-style-type: none"> • 0: E-mail message • 1: Contact • 2: Appointment
message_doctype	string(1)	S	<p>Specifies the type of email message. Valid values are:</p> <ul style="list-style-type: none"> • 0: MAPI mail message • 1: SMTP message • 2: Lotus Notes e-mail message
retention_start_date	Date	S	GMT date and time for retention start date.

3.108 Message route

Records routing information from an email message.

3.108.1 Description

Internal name: dm_message_route

This type was changed to a registered table in release 6.5, and its use is not supported for customer applications.

3.109 Message route user data

Stores route-specific user information for a particular e-mail message.

3.109.1 Description

Internal name: dm_message_route_user_data

This type was changed to a registered table in release 6.6, and its use is not supported for customer applications.

3.110 Message user data

Records application-specific information about a user referenced by an archived message.

3.110.1 Description

Internal name: dm_message_user_data

This type was changed to a registered table in release 6.6, and its use is not supported for customer applications.

3.111 Metamodel

A metamodel object records information about a Smart Container model.

3.111.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_metamodel
- Object type tag: 08

The content of a metamodel object is the XML representation the metamodel. These objects are the building blocks for Smart Container, and must never be deleted from a repository. Each metamodel object has an associated dm_logical_version aspect. Each metamodel object also has an dm_metamodel_version aspect, where *version* is the version number of the metamodel contained as content.

This object type is installed with the Smart Container DAR file during repository configuration.

3.111.2 Properties

There are no properties defined for the metamodel type. It inherits all properties from its supertype.

3.112 Method

Represents an external procedure that is invoked through the DO_METHOD function using either the DQL EXECUTE statement or the Apply method.

3.112.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_method
- Object type tag: 10

A method object represents an external procedure that is invoked through the DO_METHOD function using either the DQL EXECUTE statement or the Apply method.

3.112.2 Properties

The table describes the object properties.

Table 3-103: Method type properties

Property	Datatype	Single or repeating	Description
a_extended_property	string(32)	R	<p>Identifies the Java methods that can be run on JMS in a JMS failover setup. Valid values are:</p> <p>JMS_HA_SETUP_ENABLED is F</p> <ul style="list-style-type: none"> • JMS_LOCATION=ANY: Indicates that the method can be run on ANY JMS application server. The default value is blank. • JMS_LOCATION=ORIGINAL: Indicates that the method will run on the default local embedded JMS for current running server (JMS must be on the same host as the originating Documentum CM Server's host). • JMS_LOCATION=MAIN: Indicates if the method must be run on the Documentum CM Server bound JMS (JMS need not be on the same host as the originating Documentum CM Server's host). <p>JMS_HA_SETUP_ENABLED is T</p> <ul style="list-style-type: none"> • JMS_LOCATION=ANY: Indicates that the method can be run on ANY JMS application server. The default value is blank. • JMS_LOCATION=ORIGINAL or MAIN or LOCAL: Indicates that the method will run on the default local embedded JMS for current running server (JMS must be on the same host as the originating Documentum CM Server's host). • JMS_LOCATION=REMOTE: Indicates if the method must be run on the Documentum CM Server is not bound JMS (JMS should not be on the same host as the originating Documentum CM Server's host). • JMS_LOCATION=JMS_CONFIG_ID: Indicates if the method must be run for configured JMS only.
is_restartable	Boolean	S	Specifies whether the method is re-startable upon JMS failover. The server will not restart the method if the method fails for reasons other than a JMS failover. The default value is FALSE.

Property	Datatype	Single or repeating	Description
launch_async	Boolean	S	<p>Specifies whether the procedure runs asynchronously or not. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i>: The procedure runs asynchronously. If the method is launched on the application server, enabling SAVE_RESPONSE on the command line is ignored. • <i>F</i>: The procedure runs synchronously. <p>This property setting is ignored if the method is launched on the method server or Documentum CM Server and SAVE_RESULTS is enabled on the command line. The method is always launched synchronously.</p>
launch_direct	Boolean	S	<p>This property is used by Documentum CM Server and specifies whether the program is executed by the operating system or exec API call. This property is ignored when the method is executed on the application server or method server. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i>: The server uses the exec call to execute the procedure. The method_verb must be a fully qualified path name. • <i>F</i>: The server uses the system call to execute the procedure. <p>If SAVE_RESULTS is enabled on the command line, the server ignores the launch_direct setting and the system call executes the procedure.</p>
method_args	string(255)	R	A list of command line arguments for the procedure. Not currently used.
method_type	string(32)	S	<p>Adds -f before the file name and passes all arguments specified on the DO_METHOD command line to the program. Valid values are:</p> <ul style="list-style-type: none"> • <i>dmawk</i>: The method is executed using Documentum CM Server and the executable is stored as content. • <i>dmbasic</i>: The method is executed on the method server and the executable is stored as content. • <i>java</i>: The method is executed on an application server and the executable is stored as content.

Property	Datatype	Single or repeating	Description
method_verb	string(255)	S	<p>The command-line name of the procedure or the interpretive language name that executes the program file. Valid values are:</p> <ul style="list-style-type: none"> • <i>dmbasic</i>: The program is written in Docbasic. • <i>dmawk32</i>: The program runs on a Windows machine and is written in dmawk. • <i>dmawk</i>: The program runs on a Linux machine and is written in dmawk. • <i>Namespace</i>: The method is executing on the application server. For example, <code>com.foo.<package_name>.<class></code> or <code>fooPayMethod</code>.
run_as_server	Boolean	S	<p>Specifies whether to run the method as the server account. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i>: The method is run as the server account. Use this value to execute a method on the method server or application server. • <i>F</i>: The method is not run as the server account. <p>The default value is <i>F</i>.</p>
success_return_codes	integer	R	<p>Specifies the values for the <code>a_last_return_code</code> property of the associated job object type.</p> <p>If no value is assigned, the <code>success_return_codes</code> property is ignored.</p> <p>If the <code>a_last_return_code</code> property is not set to one of the <code>success_return_codes</code> values, the <code>dm_run_dependent_jobs</code> method assumes that the job failed.</p> <p>The <code>success_return_codes</code> property is set to 0 for replication jobs.</p>
success_status	string(255)	S	<p>Defines the value for the <code>a_current_status</code> property in a completed job.</p> <p>Without an assigned value, the server ignores the <code>success_status</code> property.</p> <p>If the <code>a_current_status</code> property value does not match the associated job object type must match the <code>success_status</code> value after the job completes, the <code>dm_run_dependent_jobs</code> method considers the job to have failed.</p> <p>For replication jobs, the <code>success_status</code> property value is set to the <i>Replicate Operation Complete</i> value.</p>

Property	Datatype	Single or repeating	Description
timeout_default	integer	S	The default method time-out value in seconds. The default value is 60 seconds. This value cannot be greater than the timeout_max value.
timeout_max	integer	S	Specifies the maximum method time-out value that can be entered on the command line. The default is 300 seconds.
timeout_min	integer	S	Specifies the minimum method time-out value that can be entered on the command line. The default value is 30 seconds. This value cannot be greater than the timeout_default value.
trace_launch	Boolean	S	<p>Specifies whether the server logs internal trace messages generated by the executing program. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i>: The trace messages are stored in the session log. • <i>F</i>: The trace messages are not logged. <p>The default value is <i>F</i>.</p> <p>If the method is executing on the application server, the trace records information about the method launch.</p>
use_method_content	Boolean	S	<p>Specifies whether the program is stored as content in the method object or as an external file. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i>: The program that is stored as the method content. The interpretative program language must be specified in the method_verb property. • <i>F</i>: The program is stored as an external file. The file path to the program or script must be specified as a namespace in the method_verb property. <p>The value is ignored if the method is executing on an application server.</p>

Property	Datatype	Single or repeating	Description
use_method_server	Boolean	S	<p>Specifies whether to use Documentum CM Server, a method server, or an application server to execute Dmbasic or Java methods. Modifying this property requires superuser privileges. Valid values are:</p> <ul style="list-style-type: none"> • T: Dmbasic or Java methods are executed on a method server or an application server. The server must be specified in the method_type value. • F: Dmbasic or Java methods are executed on Documentum CM Server.

3.113 MIP configuration

Contains information about the Microsoft Information Protection (MIP) configuration.

3.113.1 Description

- Supertype: dm_sysobject
- Subtypes: None
- Internal name: dmc_mip_config
- Object type tag: 08

The MIP configuration object contains credentials. These stored credentials are used for authentication with Azure Rights Management Service (RMS). You can create only one MIP configuration object per repository.

3.113.2 Properties

The following table describes the object properties:

Table 3-104: MIP configuration type properties

Property	Datatype	Single or repeating	Description
client_id	CHAR (255)	S	Client ID of the registered application.
tenant_id	CHAR (255)	S	Unique ID of the tenant.
client_secret	CHAR (255)	S	Client secret of the registered application.

Property	Datatype	Single or repeating	Description
is_mip_enabled	Boolean	S	Indicates if MIP is enabled.



Note: The MIP configuration type is applicable only for cloud deployment.

3.114 MIP trust configuration

Contains information about the MIP-trusted Foundation Java API clients.

3.114.1 Description

- Supertype: dm_sysobject
- Subtypes: None
- Internal name: dmc_mip_trust_client
- Object type tag: 08

The MIP trust configuration object contains information about the MIP-trusted Foundation Java API clients. Only the MIP-trusted Foundation Java API clients can operate on the MIP enabled content.

3.114.2 Properties

The following table describes the object properties:

Table 3-105: MIP trust configuration type properties

Property	Datatype	Single or repeating	Description
client_id	string (255)	S	Client ID of the Foundation Java API client instance.
hostname	string (255)	S	Name of host on which the client is installed.
is_mip_trusted_client	Boolean	S	Indicates if the Foundation Java API client is MIP-trusted.
certificate_value	string (1900)	S	Base 64-encoded DER certificate.



Note: The MIP trust configuration type is applicable only for cloud deployment.

3.115 Module

Represents a business object module.

3.115.1 Description

- Supertype: Folder
- Subtypes: Aspect Type, Validation Module
- Internal name: dmc_module
- Object type tag: 0b

A module object stores information about a business object module. The name of the module is recorded in the object_name property of the module object. The name must be unique. Module objects are stored in the repository in the /System/Modules/<module_type> directory, where <module_type> is the value specified in the a_module_type property.

A script installs the object type during Documentum CM Server installation. The *OpenText Documentum Composer* documentation contains more information about creating and managing type instances.

3.115.2 Properties

The table describes the object properties.

Table 3-106: Module type properties

Property	Datatype	Single or repeating	Description
a_bof_version	string(32)	S	The version level of the BOF module. This is defined by the module creator.
a_interfaces	string(255)	R	The fully qualified class names of the interfaces implemented by the BOF module.
a_is_privileged	Boolean	S	Specifies whether the module is a privileged module. Valid values are: <ul style="list-style-type: none"> • T: Privileged module. • F: Not a privileged module.

Property	Datatype	Single or repeating	Description
a_module_type	string(32)	S	<p>Specifies the module type. For user-defined modules, the a_module_type value is also user defined. Modules have one of the following values:</p> <ul style="list-style-type: none"> • <i>TBO</i> • <i>SBO</i> • <i>Aspect</i> <p><i>Aspect</i> type module objects are created and used internally only.</p>
a_privilege_roles	string(255)	R	A list of the roles that can use this module.
a_req_module_interfaces	string(255)	R	<p>The object IDs of the dmc_jar objects associated with the modules identified in req_module_names. The IDs at a particular index position are associated with the module at the corresponding index position in req_module_names.</p> <p>Multiple object IDs at one index position are separated by commas.</p>
contact_info	string(256)	S	<p>The contact information of a party who can provide information about this module. This value is user-defined.</p>
implementation_technology	string(32)	S	Specifies the technology used to create the module.

Property	Datatype	Single or repeating	Description
min_dfc_version	string(32)	S	Specifies the minimum Foundation Java API version level required to implement this module. The minimum value is 5.3.
module_description	string(255)	S	User-defined description of the module functionality.
primary_class	string(255)	S	The main implementation class. For .NET implementations, the value may consist of multiple comma-separated values.
req_module_names	string(64)	R	Names of modules on which this module depends.

3.116 Module config

Specifies a business object module for the workflow Timer servlet.

3.116.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmc_module_config
- Object type tag: 00

A module config object identifies a business object module for a workflow Timer servlet and stores argument values used by the servlet. A script creates the object type during Documentum CM Server installation.

3.116.2 Properties

The table describes the object properties.

Table 3-107: Module config type properties

Property	Datatype	Single or repeating	Description
id_attr_name	string(40)	R	The names of properties that have object IDs and are used by the servlet.
id_attr_value	ID	R	The object IDs of the properties stored in the corresponding index position in the id_attr_name property.
r_aspect_name	string(64)	R	The list of BOF module names.
string_attr_name	string(40)	R	The names of the properties that have string values and are used by the servlet.
string_attr_value	string(2000)	R	The string values of the properties stored in the corresponding index position in the string_attr_name property.

3.117 Mount point

A mount point object describes a mounted directory in a Documentum CM Server installation.

3.117.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_mount_point
- Object type tag: 3e

A mount point object describes a mounted directory in a Documentum CM Server installation.

3.117.2 Properties

The table describes the object properties.

Table 3-108: Mount point type properties

Property	Datatype	Single or repeating	Description
file_system_path	string(255)	S	<p>Specifies the full path location of the mount point. The path syntax must be appropriate for the machine on which the location resides.</p> <p>Refer to the caution documented after this table.</p>
host_name	string(128)	S	The host name of the machine on which this location resides.
mac_preferred_alias	string(32)	S	Specifies the preferred alias for the system to use when a Macintosh client references the mount point.
object_name	string(255)	S	<p>The name of the mount point object. Generally, the name is inherited from the SysObject supertype. It is included here because the mount point name is required for creating a mount point object.</p>
security_type	string(32)	S	<p>Specifies the security level applied to this directory location. Valid values are:</p> <ul style="list-style-type: none"> • <i>public_open</i> • <i>public</i> • <i>private</i> <p>The default value is <i>private</i>.</p>

Property	Datatype	Single or repeating	Description
unix_preferred_alias	string(32)	S	Specifies the preferred alias for the system to use when the client references the mount point. This property is not required for Linux servers.
win_preferred_alias	string(32)	S	Specifies the preferred alias for the system to use when a Windows client references the mount point.

**Caution**

Be sure that the combination of the host and path you specify does not point to the same physical location as any other file stores. If two file stores use the same physical location, data loss may result.

3.118 Network location map

Records information about the network locations of client hosts.

3.118.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_network_location_map
- Object type tag: 08

A network location object describes the network location of client sites. Instances of the object type are used to determine a location of a user relative to a Documentum CM Server.

3.118.2 Properties

The table describes the object properties.

Table 3-109: Network location map type properties

Property	Datatype	Single or repeating	Description
begin_near_ip_address	string(80)	R	The beginning of the IP address range representing IP addresses near to this network location. Each index position represents the beginning of one range. The end of that particular range is recorded in the corresponding index position in the end_near_ip_address property.
default_netloc	Boolean	S	Specifies whether this network location can serve as a default network location for a client. The default location is used when no other mapping for the IP address of a user exists.
end_near_ip_address	string(80)	R	The end of the IP address range representing IP addresses near to this network location. Each index position represents the end of one range. The beginning of that particular range is recorded in the corresponding index position in the begin_near_ip_address property.

Property	Datatype	Single or repeating	Description
ip_address_range	string(240)	R	Specifies an IPv6 address range for the network location. Either the prefix method or the IPv6address1 - IPv6address2 method can be used to indicate the range. For example, the following two examples represent the same IPv6 range: 2001:0db8:1234::/48 2001:0db8:1234::-2001:0db8:1234:ffff:ffff:ffff:ffff:ffff:ffff
netloc_display_name	string(80)	S	A user-defined display name or description of this network location.
netloc_ident	string(80)	S	A user-defined unique identifier for this network location.

3.119 NLS DD info

Records locale-dependent data dictionary information for an object type or property.

3.119.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dm_nls_dd_info
- Object type tag: 4f

An nls dd info object holds locale-dependent data dictionary information for an object type or property. NLS DD info objects are created and managed by the server. The properties in nls dd info objects are set when users add or change data dictionary information. Each nls dd info object contains the data dictionary information specific to one locale.

Not all nls dd info properties apply to both object types and properties. Some apply only to properties.

3.119.2 Properties

The **properties applicable to object type and properties** table describes the properties that are applicable to both object types and properties. The **properties only applicable to properties** table describes the nls dd info properties that are applicable only to properties.

Table 3-110: Properties applicable to object type and properties

Property	Datatype	Single or repeating	Description
comment_text	string(255)	S	User defined.
foreign_key_msgs	string(255)	R	The error message for foreign key constraint violations.
help_text	string(255)	S	User-defined help text for the property or type.
label_text	string(64)	S	The user-defined label for the property or type.
map_data_string	string(128)	R	A list of possible data values for the property.
map_description	string(255)	R	Descriptions of the data values at the corresponding index levels in map_data_string.
map_display_string	string(128)	R	The character string to display for the data value at the corresponding index level in map_data_string.
primary_key_msg	string(255)	S	The error message for primary key constraint violations.
unique_key_msgs	string(255)	R	The error message for unique key constraint violations. The message at any one index level is displayed when the constraint at the corresponding index level is violated.

Property	Datatype	Single or repeating	Description
val_constraint_msg	string(255)	R	The error message for value constraint violations. The message at any one index level is displayed when the constraint at the corresponding index level is violated.

Table 3-111: Properties applicable only to properties

Property	Datatype	Single or repeating	Description
category_name	string(64)	S	User defined.
def_value_length	integer	S	The length, in characters, of a typical value for the property. Zero indicates an unspecified (NULL) length.
format_pattern	string(64)	S	The date properties pattern to use for date interpretation.
format_pattern_tag	integer	S	Currently unused. (Currently, this is always 1.)
map_data_string	string(128)	R	The list of possible data values for the property.
map_description	string(255)	R	Descriptions of the data values at the corresponding index levels in map_data_string.
map_display_string	string(128)	R	The character string to display for the data value at the corresponding index level in map_data_string.
not_null_msg	string(255)	S	The error message for NOT NULL constraint violations.

Property	Datatype	Single or repeating	Description
parent_id	ID	S	The object ID of the aggr domain object that references the nls dd info object.

3.120 Note

Represents an annotation for a document or other SysObject.

3.120.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_note
- Object type tag: 41

A note object represents an annotation for a document or other SysObject. The content of the annotation is stored in a content file associated with the note object.

3.120.2 Properties

Note objects inherit all of their properties from their supertype, SysObject.

3.121 OTDS license config

OpenText Directory Services (OTDS) license configuration information.

3.121.1 Description

- Supertype: None
- Subtypes: None
- Internal name: dm_otds_license_config
- Object type tag: 00

A OTDS license configuration object is created when you apply the license using OTDS.

3.121.2 Properties

The table describes the object properties.

Table 3-112: OTDS license config type properties

Property	Datatype	Single or repeating	Description
otds_url	string(255)	S	URL used to connect to OTDS where the license file is uploaded.
license_keyname	string(255)	S	License key from OTDS.
business_admin_name	string(64)	S	OTDS license administrator user name.
business_admin_password	string(255)	S	OTDS license administrator password.
additional_attr_name	string(64)	R	Used to store configuration parameter names.
additional_attr_value	string(255)	R	Used to store configuration parameter values.
i_is_replica	Boolean	S	Indicates whether the object is a local replica of an object in a remote repository.
i_vstamp	integer	S	Represents the number of committed transactions that have changed an object. This value is used for versioning, as part of the locking mechanism, to ensure that one user does not overwrite the changes made by another.

3.122 Other file

Contains information about resource fork files for Macintosh.

3.122.1 Description

- Supertype: None
- Subtypes: None
- Internal name: dmi_otherfile
- Object type tag: 23

An other file object contains information about resource fork files for Macintosh.

3.122.2 Properties

The table describes the object properties.

Table 3-113: Other file type properties

Property	Datatype	Single or repeating	Description
data_ticket	integer	S	A value that specifies how the server can access the file.
i_partition	integer	S	Specifies the partition where the object resides.
protocol	integer	S	Specifies the Macintosh access protocol.

3.123 Output device

Contains a description of a printer that is accessible to repository users.

3.123.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_outputdevice
- Object type tag: 17

An output device object contains a description of a printer that is accessible to repository users.

3.123.2 Properties

The table describes the object properties.

Table 3-114: Output device type properties

Property	Datatype	Single or repeating	Description
image_format	string(64)	S	Specifies the format that the printer uses to print images.
os_name	string(32)	S	The name of the printer as specified in the /etc/printcap file.
output_command	string(128)	S	The Linux print command that is sent to the printer.
output_formats	string(128)	R	The formats supported by the device.
output_mask	string(64)	S	The job number of a queued print job, in a platform-dependent mask.
remove_command	string(128)	S	The command that the server sends to the printer to cancel a print job.
status_command	string(128)	S	The command that the server sends when the user issues a Linux lpq command.

3.124 Package

Serves as a container documents or other SysObjects moving through a workflow.

3.124.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_package
- Object type tag: 49

Workflows use package objects to deliver one or more documents or other SysObjects from the output port of a source activity to the input port of a destination activity. The server generates the packages from the activity definition properties. Users cannot create or modify package objects.

3.124.2 Properties

The table describes the object properties.

Table 3-115: Package type properties

Property	Datatype	Single or repeating	Description
i_acceptance_date	date	S	Records the date and time when a package arrived and was accepted by an input port.
i_package_oprtn	string(64)	S	Contains a value copied from the package definition upon creation of a package.
i_package_order	integer	S	Represents the order of where its associated package definition appears in the containing port definition.
r_act_seqno	integer	S	Contains the sequence number (r_act_seqno) of the activity in which the package is being handled.
r_component_chron_id	ID	R	Chronicle ID of the object identified at the corresponding index position in r_component_id.

Property	Datatype	Single or repeating	Description
r_component_id	ID	R	Object IDs of the bound objects.
r_component_name	string(80)	R	<p>Object names of the bound objects. The name at a particular index position is the name of the object identified in r_component_id at the same index position.</p> <p>Specifies whether this property is set is configurable. It is not set by default. <i>OpenText Documentum Content Management - Server Fundamentals Guide (EDCCS250400-GGD)</i> contains more information.</p>
r_note_flag	integer	R	<p>Specifies when a note was added to the package and how to handle the note in activity transitions. Valid values are:</p> <ul style="list-style-type: none"> • 0, note added prior to activity; discard at next transition • 1, note added prior to activity; retain in next transition • 2, note added during this activity; send to next activity and then discard during transition from that activity • 3, note added during this activity; retain in subsequent transitions

Property	Datatype	Single or repeating	Description
r_note_id	ID	R	Contains the ID of a dm_note object.
r_note_writer	string(255)	R	Contains the name of the user who made notes.
r_package_flag	integer	S	<p>Indicates whether the package is visible to the activity and whether it is allowed to be empty. The value is copied from the r_package_flag property in the activity object.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning the package is invisible but cannot be empty • 1, meaning the package is visible and cannot be empty • 2, meaning the package is invisible but may be empty • 3, meaning the package is visible and may be empty
r_package_label	string(32)	R	Contains a version label, if any.
r_package_name	string(16)	S	Contains the package name.
r_package_type	string(40)	S	Contains the real type name of the bound components.
r_port_name	string(16)	S	Contains the name (port_name) of the port of the activity run-time instance in which the package is being handled.

Property	Datatype	Single or repeating	Description
r_workflow_id	ID	S	Contains the object ID of the containing workflow.

3.125 Partition scheme

Defines a particular partitioning scheme.

3.125.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dm_partition_scheme
- Object type tag: 6f

A partitioning scheme is a description of how the associated types and registered tables are partitioned in the repository. You cannot directly change the values of any property, except for object_name. The administrative method PARTITION_OPERATION is used to control these objects.

3.125.2 Properties

The table describes the object properties.

Table 3-116: Partition scheme type properties

Property	Datatype	Single or repeating	Description
object_name	string(255)	S	The name of the scheme. The name must be unique among dm_partition_scheme objects.

Property	Datatype	Single or repeating	Description
owner_name	string(255)	S	The owner of the scheme object. Only a sysadmin or user with CREATE TYPE privileges can create a scheme. Only the owner or a sysadmin can modify an existing scheme. In order to apply a partition scheme to a type, either sysadmin or CREATE TYPE privilege is required. For intrinsic types, superuser privilege is required.
partition_up_range	integer	R	The right limit (excluded) of a partition range. The largest <i>i_partition</i> value of an object in this partition is equal to (partition_up_range - 1). Partition ranges are consecutive, so the lowest <i>i_partition</i> value of an object in this partition is equal to the partition_up_range of the next lower partition.
partition_file_group	string(255)	R	The file group or tablespace where the partition will be allocated.
type_list	string(27)	R	The list of type that participate in this scheme.
table_list	string(64)	R	The list of registered tables that participate in this scheme.
partition_name	string(32)	R	The name of the partition.

Property	Datatype	Single or repeating	Description
is_validated	Boolean	S	<p>Indicates if the scheme object matches the partitioning of the associated types and registered tables.</p> <p>If a db_partition operation is run, is_validated is set to FALSE until the SQL script is run that makes the changes to the underlying database.</p>

3.126 Plugin

A plugin object represents a plug-in library used by Documentum CM Server.

3.126.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dm_plugin
- Object type tag: 67

Documentum CM Server uses plug-in libraries to access external storage areas, including content-addressed storage, retention-enabled storage, and for some forms of user authentication.

3.126.2 Properties

The table describes the object properties.

Table 3-117: Plugin type properties

Property	Datatype	Single or repeating	Description
a_hw_platform	string(255)	R	Hardware architecture on which the plug-in is running. The value is user defined. For example, a value might be Intel x86 or Sun SPARC Power PC.
a_op_sys	string(255)	R	Specifies the operating system on which the plug-in is running.
a_usage	string(255)	S	Describes the purpose of the plug-in.
a_plugin_type	string(255)	S	Specifies the type of plug-in. The value is user-defined. Some example values are: batch file Bourne shell script C-shell script DLL docbasic executable java java script sl so

3.127 Policy

A policy object contains the definition of a lifecycle.

3.127.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_policy
- Object type tag: 46

Policy objects record lifecycle definitions.

3.127.2 Properties

The table describes the object properties.

Table 3-118: Policy type properties

Property	Datatype	Single or repeating	Description
action_object_id	ID	R	Object ID of the dm_procedure that implements state actions. All properties specified in the procedure must be defined in the primary type. This is unused if java_methods is T.
alias_set_ids	ID	R	List of the object IDs of the alias set objects representing acceptable alias sets for the lifecycle.
allow_attach	Boolean	R	Indicates whether a document can be attached to this state. The default value is false.
allow_demote	Boolean	R	Indicates whether the object can be demoted from this state. The default value is false.
allow_schedule	Boolean	R	Indicates whether a state transition event (such as promote, demote, suspend, or resume event) can be scheduled when this state is the current state of the SysObject. The default value is true.

Property	Datatype	Single or repeating	Description
app_validation_id	ID	S	<p>Specifies the procedure object that validates the lifecycle. If app_validation_ver is set, then app_validation_id contains the chronicle ID of the procedure object.</p> <p>This is unused if java_methods is T.</p>
app_validation_ver	string(32)	S	<p>Version of the object identified in app_validation_id.</p> <p>This is unused if java_methods is T.</p>
entry_criteria_id	ID	R	<p>Object ID of the dm_func_expr that implements the entry criteria. All properties specified in the expression must be defined in the primary type of this policy object. The source code for all expressions defined for this policy object are stored on page 0 of the policy's content, and all p-code is stored on page 1.</p>
exception_state	string(32)	R	<p>Name of the exception state, if any. Optional for normal policy states. Ignored for exception states.</p>
extension_type	string(32)	S	<p>Name of the extension type used for the lifecycle.</p>

Property	Datatype	Single or repeating	Description
i_state_no	integer	R	A 0-base internal integer number which uniquely identifies a state. Every time a new state is added, the highest number in the state chain plus one is used. This value, not the state name, is used in sysobject to record current state and resume state.
include_subtypes	Boolean	R	<p>Indicates whether all subtypes of the type specified in the corresponding position of included_type are included. Possible values are:</p> <ul style="list-style-type: none"> • false (all subtypes are NOT included) • true (all subtypes are included) <p>The property is always true when the object is created.</p>
included_type	string(27)	R	The acceptable object types. The first position stores the primary type associated with this policy object. The later positions are valid subtypes of the primary type. At least one value is required.

Property	Datatype	Single or repeating	Description
java_methods	Boolean	S	Specifies whether the entry criteria, actions, and validation methods are implemented in Java. T means they are written in Java. F means they are written in Docbasic.
r_definition_state	integer	S	<p>Status of the policy object. Possible values are:</p> <ul style="list-style-type: none"> • 0, meaning draft • 1, meaning validated • 2, meaning installed <p>The property is always 0 when the object is created.</p>

Property	Datatype	Single or repeating	Description
return_condition	integer	R	<p>Used when return_to_base is TRUE to identify which operations return an object to the policy's base state. Valid values are:</p> <ul style="list-style-type: none">• 0, return to base state on Checkin or Save• 1, return to base state on checkin• 2, return to base state on Save• 4, return to base state on Saveasnew• 8, return to base state on Branch <p>The default value is 0.</p> <p>To identify multiple operations, sum the integers. For example, if you want to return to the base state on Checkin or Saveasnew, set the property to 5.</p>

Property	Datatype	Single or repeating	Description
return_to_base	Boolean	R	<p>TRUE returns an object to the base state when the operations identified in the corresponding index position in return_condition are performed.</p> <p>The setting in each index position applies to the state in the corresponding index position in state_name.</p> <p>For a Checkin or Branch method, the affected object is the new version. For a Saveasnew, the affected object is the new copy. For a Save, the affected object is the saved object.</p> <p>The entry criteria of the base state and actions are applied to the object when it is returned to base. If the entry criteria are not met, the method fails. If the actions fail, the object remains in the base state.</p> <p>FALSE means to leave the object in the current state.</p> <p>The default value is FALSE.</p>
state_class	integer	R	Indicates whether the state is a normal state (with value 0), or an exception state (with value 1). The default value is 0.
state_description	string(128)	R	Purpose of the state.

Property	Datatype	Single or repeating	Description
state_name	string(32)	R	Name of the state. Unique within the policy definition to which the state belongs. A state name must start with a letter and cannot contain colons, periods, or commas.
state_type	string(32)	R	Name of the state type describing the state at the corresponding index position in state_name.
system_actions	ID	R	<p>Object ID of tcf activity object describing a sequence of actions through modules implementing the IDfLifecycleAction interface</p> <p>This is only set if java_methods is T.</p>
type_override_id	ID	R	<p>Object ID of dm_aggr_domain which contains the type override information prepared by data dictionary.</p> <p>This property controls properties of the document (whether properties are read-only, or hidden, and so on) when the document is in this state.</p>

Property	Datatype	Single or repeating	Description
user_action_id	ID	R	<p>Object ID of the dm_procedure that implements user-defined actions, if the user_action_ver is unspecified; chronicle ID, otherwise. All properties specified in the procedure must be defined in the primary type.</p> <p>This is unused if java_methods is T.</p>
user_action_service	string(128)	R	<p>Names of the module objects implementing the user actions.</p> <p>This is only set if java_methods is T.</p>
user_action_ver	string(32)	R	<p>The version label of user_action_id. If unspecified, the user_action_id is the id of the dm_procedure that contains user-defined actions. Otherwise, the user_action_id is the chronicle id. Together with user_action_id, this property allows users to associate a late bound user action object with a lifecycle.</p> <p>This is unused if java_methods is T.</p>

Property	Datatype	Single or repeating	Description
user_criteria_id	ID	R	<p>Object ID of the dm_procedure that implements the user-defined entry criteria for the state identified at the corresponding index position in state_name. All properties specified in the procedure must be defined in the primary type.</p> <p>This is unused if java_methods is T.</p>
user_criteria_service	string(128)	R	<p>Names of the module objects implementing the entry criteria</p> <p>This is only set if java_methods is T.</p>
user_criteria_ver	string(32)	R	<p>The version label of user_criteria_id. If specified, the specified version of the procedure identified in user_criteria_id is used. If a version label is not specified, the version identified in user_criteria_id is used. Together with user_criteria_id, this property allows users to late bind an entry procedure with a lifecycle.</p> <p>This is unused if java_methods is T.</p>

Property	Datatype	Single or repeating	Description
user_postproc_id	ID	R	<p>Object ID of the dm_procedure that implements the user-defined procedure executed after entry into the state identified in the corresponding index position in state_name.</p> <p>This property may be used in conjunction with user_postproc_ver, to late-bind a particular version of the procedure to the lifecycle state. For more information about late-binding, see <i>OpenText Documentum Content Management - Server Fundamentals Guide (EDCCS250400-GGD)</i>.</p> <p>This is unused if java_methods is T.</p>
user_postprocessing_service	string(128)	R	<p>Names of the module objects implementing the post-processing actions</p> <p>This is only set if java_methods is T.</p>

Property	Datatype	Single or repeating	Description
user_postproc_ver	string(32)	R	<p>Version labels identifying which versions of the procedures specified in user_postproc_id are to be executed.</p> <p>The label at a particular index position is applied to the procedure identified at the corresponding index position in user_postproc_id.</p> <p>Setting this property for a particular procedure allows you to late-bind a particular version of a procedure to the lifecycle state. For more information, see <i>OpenText Documentum Content Management - Server Fundamentals Guide (EDCCS250400-GGD)</i>.</p> <p>This is unused if java_methods is T.</p>
user_validation_service	string(128)	R	<p>Names of the module objects implementing user validation</p> <p>This is only set if java_methods is T.</p>

3.128 Preset info

Relates a preset to a folder.

3.128.1 Description

- Supertype: Relation
- Subtypes: None
- Internal name: dmc_preset_info
- Object type tag: 37

A preset info object relates a preset to a folder and records information about how and when the preset should be updated when changes occur to the folder. This object type is installed with the Preset DAR file during repository configuration.

3.128.2 Properties

The table describes the object properties.

Table 3-119: Preset info type properties

Property	Datatype	Single or repeating	Description
copy_source_id	ID	S	If the related folder was created by a copy operation, this property records the object ID of the source folder.
is_dirty	Boolean	S	Indicates that a change was made to the folder that requires a change to the preset of the folder.
preset_package_id	ID	S	Object ID of the preset for the folder.
preset_repository_path	string(255)	S	Specifies the location of the presets for this folder.

3.129 Preset package

3.129.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dmc_preset_package
- Object type tag: 09

A preset package object records the preset configurations for a particular scope combination. This object type is installed with the Preset DAR file during repository configuration.

3.129.2 Properties

The table describes the object properties.

Table 3-120: Preset package type properties

Property	Datatype	Single or repeating	Description
last_modifier	string(32)	S	User name of the user who last modified the object. This is the name of the actual user who initiated the change through the PresetService.
last_modifier_repository	string(32)	S	Name of the repository to which the user was connected when the preset was modified.
locked_by	string(32)	S	If a preset is checked out, this property records the name of the user who initiated the change to the preset through PresetService.
scope_key	string(32)	S	Key that represents the unique scope combination to which the preset configuration applies.

3.130 Procedure

Procedure objects store Docbasic procedures that extend the behavior of the OpenText Documentum CM clients.

3.130.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_procedure
- Object type tag: 08

Docbasic is an interpretive language that you can use to write applications to extend or customize the behavior of OpenText Documentum CM clients or Documentum CM Server. Depending on where they are stored in the repository, procedures can be executed automatically, when a user connects to a repository, or on demand, when users select a menu item. Users must have at least Read permission on the procedure object to execute it.

3.130.2 Properties

The table describes the object properties.

Table 3-121: Procedure type properties

Property	Datatype	Single or repeating	Description
user_runnable	Boolean	S	Indicates whether the procedure is ready for use.

3.131 Process

A process object contains the definition of a workflow process.

3.131.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_process
- Object type tag: 4b

A process object is created when a user creates a workflow definition. There are three inherited properties that are reserved for internal use for process objects: a_special_app, a_status, and a_application_type.

3.131.2 Properties

The table describes the object properties.

Table 3-122: Process type properties

Property	Datatype	Single or repeating	Description
act_choose_by	string(128)	R	Specifies by name the activity whose performer will choose the performer for the activity in the corresponding index position in act_choose_for.
act_choose_for	string(128)	R	Specifies by name the activity whose performer will be chosen by the performer of the activity named at the corresponding index position in act_choose_by.
act_choose_name	string(128)	R	A user-defined name that links the performers chosen by the performer of the activity in the corresponding index position in act_choose_by to other activities.
act_group_id	ID	R	Contains the object IDs of the tactility groups (dm_activity objects) defined for this process.
act_group_name	string(128)	R	Contains the activity group names. The name at each index position corresponds to the activity group identified at the corresponding index position in act_group_id.

Property	Datatype	Single or repeating	Description
act_performer_from	string(128)	R	Name of the activity from which the performer or performers of the activity in act_performer_to are selected.
act_performer_rule	integer	R	Indicates whether the activity identified in the act_performer_to property is performed by the last performer or all performers of the activity identified in the act_performer_from property. Valid values are: <ul style="list-style-type: none"> • 0: The activity is performed by the last performer identified in the act_performer_from property. • 1: The activity is performed by all the performers identified in the act_performer_from property.
act_performer_to	string(128)	R	Name of the activity whose performers are selected according to the rule defined in act_performer_rule.
calendar_id	ID	S	Reserved for future use.
current_environment	string(64)	S	Contains the deployment environment that is currently in use. The value of current_environment must be among the values defined in the deploy_environment attribute.

Property	Datatype	Single or repeating	Description
deploy_environment	string(64)	R	Contains the deployment environments defined for this process.
execution_flag	integer	S	Reserved for future use.
is_private	Boolean	S	Indicates whether the definition is private or public. The default is private.
package_control	integer	S	<p>Controls whether Documentum CM Server sets the <code>r_component_name</code> property of a package object when the component name is identified in the <code>Addpackage</code> method. Valid values are:</p> <ul style="list-style-type: none"> • 0: Allows the server to set the <code>r_component_name</code> property to the component name • 1: Requires the server to set the property to a blank value. <p>The default value is 0.</p> <p>The value in <code>package_control</code> is only effective if the docbase config property named <code>wf_package_control_enabled</code> is set to T.</p>

Property	Datatype	Single or repeating	Description
parameter_name	string(255)	R	Contains the parameter names. The name at each index position corresponds to the parameter identified at the corresponding index position in parameter_id.
parameter_id	ID	R	Contains the object IDs of the parameters (dmc_process_parameter objects) defined for this process.
perf_alias_set_id	ID	S	Contains the object ID of the alias set object used to resolve performer aliases when a workflow based on this process definition is created.
r_act_def_id	ID	R	Contains the object IDs of the activities (dm_activity objects) defined for this process.
r_act_name	string(128)	R	Contains the activity names. The name at each index position corresponds to the activity identified at the corresponding index position in r_act_def_id.
r_act_priority	integer	R	Indicates the importance of the activity compared to other activities in this process. The priority at each index position is assigned to the activity defined in r_act_def_id at the corresponding index position. The default value is 0.

Property	Datatype	Single or repeating	Description
r_act_type	integer	R	<p>Indicates the type of the activity identified at the corresponding index position in r_act_def_id. Valid values are:</p> <ul style="list-style-type: none"> • 0, for step (default) • 1, for begin • 2, for end • 3, for exception • 4, for initiate
r_definition_state	integer	S	<p>Indicates the state of the process definition. Valid values are:</p> <ul style="list-style-type: none"> • 0, for draft • 1, for validated • 2, for installed
r_link_dest_act	string(128)	R	Specifies the destination activity for the link.
r_link_dest_port	string(16)	R	Indicates the input port of the destination activity.
r_link_name	string(16)	R	Uniquely identifies the link.
r_link_src_act	string(128)	R	Specifies the source activity for the link.
r_link_src_port	string(16)	R	Indicates the output port of the source activity.

Property	Datatype	Single or repeating	Description
sd_element_acl	ID	R	Object IDs of the ACLs associated with the structured data element objects for this process. The ACL identified at each index position is associated with the structured data element instance identified at the corresponding index position in sd_element_name.
sd_element_default_acl	ID	S	The default ACL used for an structured data element when there is no ACL specified in sd_element_id for the element.
sd_element_default_value	string(2000)	R	<p>Default value for simple structured date element types. The value in this property at a particular index position corresponds the type identified in the corresponding position in the sd_element_type property.</p> <p>If the type is a simple type but is not a string datatype, the value is converted to a string before being stored in this property.</p> <p>If the type is not a simple type, the value in this property is an empty string.</p>

Property	Datatype	Single or repeating	Description
sd_element_name	string(255)	R	Names of the structured data elements. Each name must be unique among the names of other structured data elements and packages within the process.
sd_element_options	integer	R	<p>Indicates whether to remove the runtime instance of the structured data element when the workflow is finished. Valid values are:</p> <ul style="list-style-type: none"> • 0, remove the structured data element instance (default). • 1, keep the instance. • The value at each index position is associated with the structured data element instance identified at the corresponding index position in sd_element_name .
sd_element_parent_id	ID	R	<p>List of the object IDs of the wfsd element parent objects that serve as the parents of structured data elements for this process.</p> <p>The object ID at each index position is the parent for the structured data element at the corresponding index position in workflow definition.</p>

Property	Datatype	Single or repeating	Description
sd_element_type	ID	R	Object IDs of the wfsd type info objects that identify the datatypes of the structured data element. The ID at each index position is associated with the structured data element instance identified at the corresponding index position in sd_element_name.
system_name	string (512)	S	System name.
target_namespace_uri	string(2000)	S	Contains the target namespace uri used by the inbound web-service activities contained in the process.

3.132 Process correlation set

Stores information about a correlation set.

3.132.1 Description

- Supertype: Relation
- Subtypes: None
- Internal name: dmc_process_correlation_set
- Object type tag: 37

A process correlation set object records information about a correlation set defined for a workflow definition. A correlation set defines what data will be used to communicate between the OpenText™ Documentum™ Content Management Process Engine and external applications at runtime. The object type is installed by a script when Documentum CM Server is installed.

3.132.2 Properties

The table describes the object type properties.

Table 3-123: Process correlation set type properties

Property	Datatype	Single or repeating	Description
correlation_set_alias	string(255)	R	Alias names used for this correlation set. The name can be either a valid package name or a structured data element defined for the process.
correlation_set_attr	string(2000)	R	Names of the properties referenced in the correlation set. The property specified at a particular index position must be referenced by the correlation set identified in the corresponding index position in correlation_set_alias.

3.133 Process parameter

Specifies a parameter object of a process definition.

3.133.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_process_parameter
- Object type tag: 08

The process parameter object contains the parameters associated with process objects (dm_process).

3.133.2 Properties

The table describes the object properties.

Table 3-124: Process parameter type properties

Property	Datatype	Single or repeating	Description
boolean_value	Boolean	S	Value of the parameter if the data type identified in parameter_type is boolean.
date_value	Date	S	Value of the parameter if the data type identified in parameter_type is date.
deploy_environment	string(64)	S	Deployment environments of the parameter.
double_value	double	S	Value of the parameter if the data type identified in parameter_type is double.
int_value	integer	S	Value of the parameter if the data type identified in parameter_type is int.
parameter_description	string(255)	S	Description of the parameter.
parameter_name	string(255)	S	Name of the parameter.
parameter_type	integer	S	Data type of the parameter. Valid values are: <ul style="list-style-type: none"> • 0: Boolean • 1: Int • 2: String • 3: ID (not used) • 4: Date • 5: Double

Property	Datatype	Single or repeating	Description
process_id	ID	S	ID of the process (dm_process object) for which the parameter is defined.
string_value	string(2000)	S	<p>Value of the parameter if the data type identified in parameter_type is string.</p> <p>The value at each index position corresponds to the deploy environment identified at the corresponding index position in deploy_environment. This is not set unless the datatype is string.</p>

3.134 Public key certificate

Used to decrypt instructions encrypted using a private cryptographic key.

3.134.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_public_key_certificate
- Object type tag: 08

A public key certificate object stores a public key certificate. A public key certificate is used by Accelerated Content Services and Branch Office Caching Services servers to decrypt the instructions sent by Documentum CM Server in response to a request for content from an Accelerated Content Services or Branch Office Caching Services server. A public key certificate is also used to authenticate Foundation Java API instances.

Public key certificate instances are created automatically by Documentum CM Server. Public key certificate objects are stored in the /System/Cryptographic/Keys folder. By default, public key is protected by an internal ACL, where dm_world has READ permission, owner and dm_superusers have delete permission.



Caution

Public key certificate objects are for internal use only. Do not modify, remove, or add these objects.

3.134.2 Properties

The table describes the object properties.

Table 3-125: Public key certificate type properties

Property	Datatype	Single or repeating	Description
certificate_value	string(1900)	S	Base 64-encoded DER certificate.
key_identifier	string(40)	S	Base 64-encoded SHA1 digest of the DER public key.
key_type	integer	S	Specifies the key that is used by Documentum CM Server. Valid values are: <ul style="list-style-type: none"> • 1: Accelerated Content Services encryption • 2: Foundation Java API authentication • 3: Branch Office Caching Services authentication
private_key_identifier	string(40)	S	Base 64-encoded SHA1 digest of the DER public key value in the dm_cryptographic_key.key_identifier property. This property is empty if the key_value is 2.

3.135 Qual comp

A qual comp object represents a component routine.

3.135.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_qual_comp
- Object type tag: 08

Component routines are written in a language compatible with the platform on which they are executed. They are used by client applications.

3.135.2 Properties

The table describes the object properties.

Table 3-126: Qual comp type properties

Property	Datatype	Single or repeating	Description
available_offline	Boolean	S	Currently unused.
class_name	string(64)	S	Specifies the functionality provided by the component; for example, checkin or checkout.

Property	Datatype	Single or repeating	Description
client_capability	integer	S	<p>Indicates who can use the component. Valid values are:</p> <ul style="list-style-type: none"> • 0: No one • 1: Consumer • 2: Contributor • 3: Consumer and contributor • 4: Coordinator • 5: Coordinator and consumer • 6: Coordinator and contributor • 7: Consumer, contributor, and coordinator • 8: Administrator • 9: Administrator and consumer • 10: Administrator and contributor • 11: Administrator, contributor, and consumer • 12: Administrator and coordinator • 13: Administrator, coordinator, and consumer • 14: Contributor, coordinator, and administrator • 15: All users <p>If there is no value specified for this property, any user can use the component.</p>
component_obj_id	ID	S	Object ID of the component object associated with the qual comp object.

Property	Datatype	Single or repeating	Description
is_global	Boolean	S	TRUE indicates that the component is executable at all times. The user is not required to select an object before executing the component. FALSE indicates that there must be a currently selected object to execute the component.
nls_label	string(255)	S	Currently unused.
valid_groups	string(255)	R	Names of groups whose users can use this component. If there are no values, any user may use this component. If there are one or more groups named, users must belong to one of the groups to use the component.

3.136 Query

A query object stores a DQL query as content.

3.136.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_query
- Object type tag: 0a

Query objects are used by Desktop Client. Their content files are saved DQL queries that a user can execute by selecting them from a menu. Users can create query objects in Desktop Client or through the API. To execute a saved query, a user must have at least Read permission on the query object.

3.136.2 Properties

The Query type has no defined properties. It inherits all of its properties from its SysObject supertype.

3.137 Queue item

A queue item object stores information about an object placed on a queue of user.

3.137.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_queue_item
- Object type tag: 1b

A queue item object records information about tasks and other objects queued to the Inbox of a specific user. Tasks are typically queued to a user when that user is selected as the performer of a workflow task. Users and applications can query queue item objects directly.

3.137.2 Properties

The table describes the object properties.

Table 3-127: Queue item type properties

Property	Datatype	Single and repeating	Description
a_content_type	string(32)	S	File format to use when opening the object associated with the task from the inbox.
a_operations	string(16)	S	User-defined.
actual_start_date	Date	S	Actual date that the task is started. This is set when the task is acquired.
content_type	string(32)	S	Not used for work items.
date_sent	Date	S	The date the object was sent.

Property	Datatype	Single and repeating	Description
delete_flag	Boolean	S	<p>Indicates whether the peer work item has been completed or if the event has been delivered or processed. Valid values are:</p> <ul style="list-style-type: none">• <i>F</i>: Not completed or delivered.• <i>T</i>: Completed or processed.

Property	Datatype	Single and repeating	Description
dependency_type	string(10)	S	<p>Records the circumstances of the work item's creation. Valid values are:</p> <ul style="list-style-type: none"> • 0: Normal. Standard work item creation (default) • 1: Revert. The item was created by reverting a previous task. • 2: A resolution error occurred while trying to resolve an alias for a normal activity. • 3: Revert and resolution error. The item was reverted and resolution error occurred for revert activity. • 4: Delegated. The item was created when the performer of the previous task delegated the task or when Documentum CM Server automatically delegates the task to the user identified in the performer's delegated_user property. • 5: Repeated. The item was created when the activity performer chose to repeat the activity, with another performer.

Property	Datatype	Single and repeating	Description
			<ul style="list-style-type: none"> • 6: Failed auto_delegation. The performer does not exist, the performer is not available and has no indicated replacement, the indicated replacement does not exist, or the replacement availability is disabled.
dequeued_by	string(255)	S	Contains the name of the user who completes the peer work item or the name of the user who dequeued the item, either explicitly or implicitly.
dequeued_date	Date	S	Contains the date and time that the item was dequeued or when an event was delivered and processed.
due_date	Date	S	Date by which the task or work item should be completed.
event	string(64)	S	<p>A system- or user-defined event.</p> <p>If dependency_type is 4 and the task was delegated by the performer, this is set to dm_delegatedworkitem. If the dependency_type is 4 and the task was automatically delegated by the server, the event is dm_startedworkitem.</p>

Property	Datatype	Single and repeating	Description
event_detail	string(2000)	S	For dm_batch_flush events, this attribute contains a list of IDs created or deleted in each batch flush. The list is formatted as objID1(op), objID2(op), ... where op is C for creation and D for deletion.
i_event_flags	integer	S	Used internally.
i_partition	integer	S	Specifies the partition where the object resides.
index_latency_in_seconds	integer	S	User-defined. Input is used to get oldest queue item and to always look back x seconds (in index_latency_in_seconds) from last_exec_date when QBS job run the subscription next time. QBS will not execute the DQL if input is provided. For best performance on Oracle database, add a new index on date_sent,name,task_state (in this order) on table dmi_queue_item_s.
instruction_page	integer	S	Not used for work items.

Property	Datatype	Single and repeating	Description
item_id	ID	S	<p>For a workflow task, the object ID of the peer work item.</p> <p>For events, the object ID of the object that generated the event. For checkin events, this is the object ID of the new version. For events queued to workflows, this is the workflow ID.</p> <p>If the queue item was generated by an Archive method, this is set to the object ID of the object selected by the method's predicate. If the predicate selects multiple objects, item_id is set to NULL.</p>
item_name	string(255)	S	<p>For a non-workflow event, the object name of the object that generated the event.</p> <p>For a workflow event, the object name of the process or workflow object that generated the event.</p> <p>Unused For workflow tasks or workflow-related events.</p>
item_type	string(32)	S	<p>For a workflow task, the object type of the peer work item.</p> <p>For an event, the object type of the object that generated the event.</p>

Property	Datatype	Single and repeating	Description
message	string(255)	S	Message from the task's sender to the person receiving the queued item.
name	string(255)	S	Contains the name of the user to whom the queued item was sent.
next_tasks_type	string(10)	S	Obsolete. Not used for work items.
plan_start_date	Date	S	Obsolete. Not used for work items.
position	integer	S	If the queue item represents a work item for a work queue and the package associated with the work item has a defined skill level, this property is set to the integer value of that skill level.
priority	integer	S	User or application-interpreted value that defines the priority of item.
queueperson	string(255)	S	Name of the fulltext index user. The default value is set to dm_fulltext_index_user. The value need not be changed unless queue user of xPlore IA on the repository is different from dm_fulltext_index_user and has different registered events for different types. For example, user need to change it if it is a dual mode setup.
read_flag	Boolean	S	Not used for work items.

Property	Datatype	Single and repeating	Description
remote_pending	Boolean	S	Indicates whether the queue item must be distributed to other repositories. TRUE means that the object associated with the queue item must be sent to the repository defined in target_docbase.
router_id	ID	S	Object ID of the workflow that contains the queued task.
sent_by	string(255)	S	Name of the user that sent the queued item. This is the user's Documentum CM Server user name.
sign_off_date	Date	S	Not used for work items.
sign_off_required	Boolean	S	Indicates if the task's user must sign off the task before forwarding the task.
sign_off_user	string(255)	S	Not used for work items.
source_docbase	string(120)	S	Name of the repository in which the queue item originated. This property has no value if the queue item is not a distributed item.
source_event	ID	S	Object ID of the queue item in the source repository.
source_stamp	integer	S	The i_vstamp value of the source queue item object.

Property	Datatype	Single and repeating	Description
stamp	ID	S	Object ID of the queue item object If the queue item object is a replica, this is the object ID of the replica.
supervisor_name	string(255)	S	Name of the supervisor of workflow.
target_docbase	string(120)	S	Name of the repository to which the queue item is sent. This property has no value if the queue item is not a distributed item.
task_name	string(128)	S	Contains the string event, if this is an event, or contains the activity identifier if this is a work item.
task_number	string(10)	S	For work items, the sequence number within the workflow of the activity that generated the work item.

Property	Datatype	Single and repeating	Description
task_state	string(10)	S	<p>Indicates the status of the event, work item, or object queued for indexing.</p> <p>The status of the event or work item. One of: acquired, paused, or finished.</p> <p>If the queue item represents an object queued for indexing, the valid values are:</p> <ul style="list-style-type: none">• <i>Blank</i>: The item has been queued but not yet picked up by the index agent.• <i>Acquired</i>: The index agent has picked up the object but not yet completed processing.• <i>Failed</i>: The item failed indexing and will not be available for searching.• <i>Warning</i>: Content indexing failed, but the metadata was indexed.• <i>Done</i>: Indexing of content and metadata was successful.

Property	Datatype	Single and repeating	Description
task_subject	string(512)	S	<p>For workflow events, constructed from the value in dm_activity.task_subject.</p> <p>For dm_save events, this records the object IDs of the old and new storage areas if the object is saved with a change to a_storage_type.</p> <p>For dm_move_content events, this records the object IDs of the object and the old and new storage areas.</p> <p>For dm_destroy events, this records the object ID of the object's storage area.</p>
task_type	string(10)	S	Not used for work items.
verbose	Boolean	S	<p>QBS job report will print more information for debug purpose when ON. Valid values are:</p> <ul style="list-style-type: none"> • <i>F</i>: Turned off. • <i>T</i>: Turned on.

3.138 Reference

A reference object contains the information that links an object in a remote repository to the local object that mirrors the remote object.

3.138.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dm_reference
- Object type tag: 47

Users never work directly with reference objects. Documentum CM Server or client applications create them when users perform operations on remote objects.

3.138.2 Properties

The table describes the object properties.

Table 3-128: Reference type object properties

Property	Datatype	Single or repeating	Description
binding_condition	string(32)	S	<p>Specifies which version of the remote object to fetch when the object is accessed through its reference object. Valid values are:</p> <ul style="list-style-type: none">• <i>EXACT_ID</i>: Directs the server to fetch the version that has the object ID specified in <code>reference_by_id</code>.• <i>VERSION_LABEL</i>: Directs the server to fetch the version carrying the version label defined in <code>binding_label</code>. <p><i>VERSION_LABEL</i> is the default.</p>

Property	Datatype	Single or repeating	Description
binding_label	string(32)	S	<p>Specifies the version to fetch for operations on the remote object when the value of binding_condition is VERSION_LABEL.</p> <p>The default version label is CURRENT.</p>
i_external_rep	Boolean	S	Indicates whether the replication mode was external. TRUE indicates that external replication was used.
i_global_vstamp	integer	S	Version stamp of the remote object.
i_null_ref_flag	integer	S	<p>Indicates the status of the remote object. Valid values are:</p> <ul style="list-style-type: none"> • 0: The object exists. • 1: The object does not exist. • 2: The object exists but no version matches the binding condition.
i_ref_acl_domain	string(255)	S	Owner of the ACL of the remote object.
i_ref_acl_name	string(32)	S	Name of the ACL of the remote object.
i_ref_acl_vstsamp	integer	S	Version stamp of ACL of the remote object.
i_replica_job_id	ID	S	<p>Object ID of the replication job that created the replica.</p> <p>This only has a value if the reference object is the result of a replication job.</p>

Property	Datatype	Single or repeating	Description
i_replica_master	ID	S	Object ID of the original (non-replica) object.
i_replica_source	ID	S	Object ID of the replicated object. This value will be the same as that in i_replica_master unless the replicated object was itself a replica.
i_usage_count	integer	S	The number of system-created reference objects.
local_folder_link	string(200)	S	Specifies the local folder to which the mirror object is linked. If unspecified, this defaults to the default folder of the user.
r_last_refresh	DATE	S	The time of last refresh of the mirror object.
r_mirror_object_id	ID	S	Object ID of the mirror object associated with the reference object.
r_next_refresh	DATE	S	Next scheduled refresh time for the mirror object.
r_ref_creator	string(255)	S	Name of the creator of the reference object.
r_ref_object_id	ID	S	Object ID of the actual remote object to which the reference object points.
r_ref_version_id	ID	S	Object ID of the exact object that last satisfied the binding condition.

Property	Datatype	Single or repeating	Description
r_reference_info	string(32)	S	Indicates how the reference is used. Valid values are: LINK_REFERENCE, CHECKOUT_REFERENCE, WORKFLOW_REFERENCE, ASSEMBLY_REFERENCE, ANNOTATION_REFERENCE REPLICA.
r_refresh_status	string(255)	S	Status of the last mirror object refresh.
reference_db_name	string(120)	S	Name of the repository that contains the remote object. This property must be defined if reference_by_name is used to define the version tree containing the remote object. The property is optional if reference_by_id is used to define the object's version tree.

Property	Datatype	Single or repeating	Description
reference_by_id	ID	S	<p>Object ID of the remote object.</p> <p>This is used to identify the version tree containing the remote object.</p> <p>If the value of binding_condition is EXACT_ID, then this identifies the exact object on the tree to which the reference points.</p> <p>If binding_condition is VERSION_LABEL, then this value only identifies the version tree containing the version specified in binding_label.</p> <p>If reference_by_id is set, you cannot set reference_by_name.</p>
reference_by_name	string(255)	S	<p>Full folder path of the remote object.</p> <p>If you set reference_by_name, you must also set reference_db_name.</p> <p>Reference_by_name and reference_by_id are mutually exclusive. You cannot set both.</p>
refresh_interval	integer	S	<p>Defines how often to poll the repository to determine if the mirror object needs to be refreshed. The value is specified in minutes.</p> <p>The default is one day.</p>

3.139 Registered

A registered object contains information about an underlying RDBMS table that has been registered with Documentum CM Server.

3.139.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_registered
- Object type tag: 19

The information includes the names and datatypes of some or all of the columns in the table and a list of those columns belonging to indexes. To create a registered object, you use the DQL REGISTER statement. To destroy a registered object, use the UNREGISTER statement.



Note: Registered objects are not updated automatically if the definition of their corresponding RDBMS table changes.

3.139.2 Properties

The table describes the object properties.

Table 3-129: Registered type properties

Property	Datatype	Single or repeating	Description
column_count	integer	S	Number of columns in the table.
column_datatype	string(64)	R	List of the datatypes of the columns.
column_length	integer	R	Lengths of the columns that have a string data type.
column_name	string(64)	R	List of the names of the columns in the table. The names must consist of ASCII characters.
group_table_permit	integer	S	Defines the RDBMS table permit level assigned to the registered table's group.

Property	Datatype	Single or repeating	Description
is_key	Boolean	R	Indicates if an index is built on the column.
owner_table_permit	integer	S	Defines the RDBMS table permit level assigned to the registered table's owner.
synonym_for	string(254)	S	Name of the table in the underlying RDBMS (can be an Oracle table synonym, or an MS SQL Server table alias).
table_name	string(64)	S	Name of the table. The name must consist of ASCII characters.
table_owner	string(255)	S	Name of the owner of the RDBMS table (the person who created the RDBMS table).
world_table_permit	integer	S	Defines the RDBMS table permit level assigned to the world.

3.140 Registry

A registry object contains information about a registered event.

3.140.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_registry
- Object type tag: 26

Users can register to receive notification of an event's occurrence. Users can also initiate auditing of events. Both actions generate registry objects.

3.140.2 Properties

The table describes the object properties.

Table 3-130: Registry type properties

Property	Datatype	Single or repeating	Description
a_authentication	integer	S	<p>Defines whether an application should authenticate the user before the event. Valid values are:</p> <ul style="list-style-type: none"> • 0: Do not authenticate. • 1: Authentication required. <p>This setting must be enforced by the application. Documentum CM Server ignores this property value.</p>
a_esignature_required	integer	S	<p>Specifies whether the event requires an electronic signature created with Addesignature. Valid values are:</p> <ul style="list-style-type: none"> • 0: No signature is required. • 1: A Signature is required. <p>The default value is 0.</p> <p>This property value is not used by Documentum CM Server. It is for use by client applications.</p>
audit_attr_names	string(40)	R	Names of the properties to store in the attribute_list property of the audit trail entry.

Property	Datatype	Single or repeating	Description
audit_subtypes	Boolean	S	Specifies whether to audit subtypes of the audited object type. The audit_subtypes setting is used only if the registered_id property identifies an object type.
controlling_app	string(32)	S	Specifies a particular application. If set, only objects controlled by that application are audited.
event	string(64)	S	Name of the event.
message	string(255)	S	For notifications, defines a message to be sent to the user with the notification. For events, defines a user-friendly event name.
oneshot	Boolean	S	Currently unused.
policy_id	ID	S	Chronicle ID of a lifecycle. If set, only objects attached to the lifecycle are audited.
policy_state	string(32)	S	Name of a particular state in the lifecycle identified in policy_id. If set, only objects in that state are audited.
priority	integer	S	The event priority level. This is user-defined.

Property	Datatype	Single or repeating	Description
is_audittrail	Boolean	S	Defines whether this is a notification or audit trail event. Valid values are 0, for a notification event, and 1, for an audit event.
registered_id	ID	S	Specifies the object or object type being audited or for which a user has registered interest. For individual objects, this is the chronicle ID of the object. For object types, this is the object ID of the type's dm_type object.
sendmail	Boolean	S	Indicates whether to send email notification of the event.
sign_audit	Boolean	S	Specifies whether Documentum CM Server must sign entries generated from this registration.
user_name	string(255)	S	Name of the user that registered for the event or initiated auditing.

3.141 Relation

Describes a relationship between two objects.

3.141.1 Description

- Supertype: Persistent Object
- Subtypes: Category Assign, Aspect Relation, State Extension, Validation Relation, DSM Sect Doc Attributes, DSM Doc Properties, DSM Study Report
- Internal name: dm_relation
- Object type tag: 37

A relation object describes a relationship between two objects. The properties identify the objects and some behavioral characteristics of the relationship.

3.141.2 Properties

The table describes the object properties.

Table 3-131: Relation type properties

Property	Datatype	Single or repeating	Description
child_id	ID	S	Specifies the ID of the child object.
child_label	string(32)	S	Version label of the specified child_id. This property is optional. If set, the child_id must be the chronicle ID for the child.
Description	string(255)	S	Not system defined. Provided for the convenience of the user.
effective_date	date	S	Not system defined. Provided for the convenience of the user.
expiration_date	date	S	Not system defined. Provided for the convenience of the user.
i_partition	integer	S	Specifies the partition where the object resides.

Property	Datatype	Single or repeating	Description
order_no	integer	S	Not system defined. Provided for the convenience of the user. For example, this could be used to order a set of relationships.
parent_id	ID	S	Specifies the object which is the parent in the relationship.
permanent_link	Boolean	S	Specifies whether to maintain the relationship across versions of the parent object. If the relation is created by a 5.3 Foundation Java API, this defaults to the value of dm_relation_type.permanent_link. If the relation is created by the DMCL or a pre-5.3 Foundation Java API, the default is FALSE. This property is deprecated in version 5.3.
relation_name	string(32)	S	Specifies a valid relation type object, which defines the type of relationship existing between the two objects.

3.142 Relation SSA policy

Describes the relationship between a SysObject object and a storage policy.

3.142.1 Description

- Supertype: Relation
- Subtypes: None
- Internal name: dm_relation_ssa_policy
- Object type tag: 37

A relation ssa policy object associates an ssa policy object, representing a content assignment policy, with an object type. The object type must be a SysObject or SysObject subtype. Relationships between object types and ssa policy objects are created through Documentum Administrator.

3.142.2 Properties

The table describes the object properties.

Table 3-132: Relation SSA policy type properties

Property	Datatype	Single or repeating	Description
target_object_type	string(32)	S	Specifies the object type of the related object.
ssa_policy_id	ID	S	The object ID of the ssa policy object enforced for the related object.

3.143 Relation type

Describes a relationship that can exist between two objects in the repository.

3.143.1 Description

- Supertype: Persistent Object
- Subtypes: dm_foreign_key
- Internal name: dm_relation_type
- Object type tag: 38

A relation type object describes a relationship that can exist between two objects in the repository.

When a user sets up a relationship between two objects, for example, by creating an annotation and attaching it to a document, the server creates an object of type dm_relation. The relation object links the annotation (child) to the document (parent) and describes the relationship between them by referring to the associated dm_relation_type object. A user must have Sysadmin or Superuser privileges to create or destroy a relation type object.

[“Relationships” on page 30](#) contains information about defining relationships or relationships between two objects.

3.143.2 Properties

Table 3-133: Relation type object type properties

Property	Datatype	Single or repeating	Description
a_controlling_kind	string(32)	S	Set to a value to indicate whether relationships of this type are created by users or internally. The setting is used by OpenText Documentum CM clients to determine whether to allow users to create and destroy instances of the relation type.
child_parent_label	string(255)	R	User-defined label for a child-to-parent relationship.
child_type	string(32)	S	The object type of valid child objects in the relationship.

Property	Datatype	Single or repeating	Description
copy_child	integer	S	<p>Specifies whether to copy the child in a relationship when permanent_link is T and the parent is copied. Valid values are:</p> <ul style="list-style-type: none"> • 0: Do not copy the child. • 1: Copy the child.
Description	string(250)	S	User-defined description of the relationship.
direction_kind	integer	S	<p>The direction of the relationship. Valid values are:</p> <ul style="list-style-type: none"> • 0: From parent to child. • 1: From child to parent. • 2: Bidirectional - objects are treated as siblings. <p>The default value is 0.</p>
i_partition	integer	S	Specifies the partition where the object resides.
integrity_kind	integer	S	<p>Indicates how referential integrity is enforced. Valid values are:</p> <ul style="list-style-type: none"> • 0: Allow delete. • 1: Restrict delete. • 2: Cascade delete. <p>The default value is 0.</p>
parent_child_label	string(255)	R	User-defined label for a parent-to-child relationship.
parent_type	string(32)	S	Defines the object type of a valid parent object in the relationship.

Property	Datatype	Single or repeating	Description
permanent_link	Boolean	S	<p>Specifies whether the relationship is maintained when the parent is copied or versioned. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i>: The relationship is maintained with the new parent object. • <i>F</i>: The relationship is not maintained. <p>The default value is <i>F</i>.</p>
relation_name	string(32)	S	The name of the relationship. Names must be unique. The names of system-defined relationships begin with dm_.

Property	Datatype	Single or repeating	Description
security_type	string(10)	S	<p>Defines the security applied to objects of type dm_relation whose relation_name property value matches the relation_name property for the dm_relation_type object. Valid values are:</p> <ul style="list-style-type: none">• <i>SYSTEM</i>: Only users with superuser or sysadmin privileges can create, modify, or drop dm_relation objects having this relation_name.• <i>PARENT</i>: Security is determined by the object type of the parent object.• <i>CHILD</i>: Security is determined by the object type of the child object.• <i>NONE</i>: No security is applied to the dm_relation objects representing instances of this relationship.

3.144 Relationship def

3.144.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_relationship_def
- Object type tag: 08

A relationship def object records the definition of a relationship. These objects are created using the IDfRelationshipInfo interface in Foundation Java API. Each relationship def object must have a binding aspect attached that describes the specific physical binding that characterizes the relationship.

The object type is installed with the Smart Container DAR file when a repository is configured.

3.144.2 Properties

The table describes the object properties.

Table 3-134: Relationship def type properties

Property	Datatype	Single or repeating	Description
is_composition	Boolean	S	<p>Indicates whether the relationship is a strict composition, meaning the related children have a single parent and that destroying the parent also destroys the children. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i>: The relationship is a strict composition. • <i>F</i>: It is not a strict composition. <p>The default value is <i>F</i>.</p>

Property	Datatype	Single or repeating	Description
is_private	Boolean	S	<p>Indicates whether the relationship is available for external use. Valid values are:</p> <ul style="list-style-type: none"> • <i>T</i>: The relationship is private, not available for public use. • <i>F</i>: The relationship may be used by external applications.
operation_name	string(255)	R	<p>Name of an operation to be applied after some triggering event. The name at any particular index position is associated with the module and metadata in the corresponding index positions in operation_module and operation_module metadata.</p> <p>The name should be a fully qualified name, such as a Java package name, to guarantee uniqueness.</p>
operation_module	string(255)	R	<p>Module that implements the operation identified in operation_name. The module at a particular index position is the module that implements the operation identified at the same index position in operation_name.</p>

Property	Datatype	Single or repeating	Description
operation_module_metadata	string(1024)	R	Metadata used to configure the operation module. The metadata at a particular index position is the metadata used to configure the module identified at the same index position in operation_module.
rel_description	string(1024)	S	User-defined description of the relationship.
rel_degree	integer	S	Indicates the degree of the relationship from parent to child. Valid values are: <ul style="list-style-type: none"> • 0: One-to-one relationship • 1: Many-to-one relationship • 2: Many-to-many relationship
required_source_aspects	string(64)	R	List of aspects that will be associated with an instance of the source (parent) type when participating in this relationship.
required_target_aspects	string(64)	R	List of aspects that will be associated with an instance of the target (child) type when participating in this relationship.
source_constraint_set_id	ID	S	Object ID of the dmc_constraint_set object that records the set of constraints applied to the source object in the relationship.

Property	Datatype	Single or repeating	Description
source_role	string(255)	S	Name for the role played by the source object in the relationship. For example, if the relationship relates a customer with a vendor, the role for the customer may 'buyer'.
source_type	string(27)	S	Specifies the top-most type in the hierarchy of types that may participate as parent in this relationship. This type or any of its subtypes may be a parent in the relationship.
target_constraint_set_id	ID	S	Object ID of the dmc_constraint_set object that records the set of constraints applied to the target object in the relationship.
target_role	string(255)	S	Name for the role played by the target object in the relationship. For example, if the relationship relates a customer with a vendor, the role for the vendor may 'seller'.
target_type	string(27)	S	Specifies the top-most type in the hierarchy of types that may participate as child in this relationship. This type or any of its subtypes may be a child in the relationship.

This table describes the default aspect properties for the object type.

Table 3-135: Relationship def type default aspect properties

Property	Datatype	Single or repeating	Description
logical_version	string(32)	S	Supports logical versioning of persistent objects From the aspect named dm_logical_aspect
annotation_name	string(255)	R	A non-qualifiable property that stores the names of the annotation From the aspect named dm_annotationable
module_name	string(255)	R	A non-qualifiable property that stores module names From the aspect named dm_annotationable
annotation_namemetadata	string(1054)	R	A non-qualifiable property that stores annotation metadata values From the aspect named dm_annotationable

3.145 Replica record

Contains information about replicated content in distributed storage areas.

3.145.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_replica_record
- Object type tag: 2d

A replica record object contains information about replicated content in distributed storage areas. The object is created and used by Documentum CM Server to manage distributed content storage areas.

3.145.2 Properties

The table describes the object properties.

Table 3-136: Replica record type properties

Property	Datatype	Single or repeating	Description
component_id	ID	R	Object IDs of the components storage areas that contain replicas of the content file.
data_ticket	integer	R	Information used internally to fetch and save the content.
epoch_number	integer	S	Used internally to manage distributed storage areas.
format_id	ID	S	Object ID of the format object representing the format of the content file associated with the replica record.
i_partition	integer	S	Specifies the partition where the object resides.
other_ticket	integer	R	Information used internally to fetch and save the content.
owner_id	ID	S	Object ID of the distributed storage area that contains the content file.
r_storage_ticket	integer	S	Used internally to manage the content

3.146 REST store

Represents a REST content storage area.

3.146.1 Description

- Supertype: Store
- Subtypes: none
- Internal name: dm_rest_store
- Object type tag: inherited from Store

REST store object represents an external content storage area which supports REST APIs. The properties in REST store objects may be changed only by users with Sysadmin or Superuser user privileges.

3.146.2 Properties

REST store inherit all of the properties from the Store supertype.

Table 3-137: REST store properties

Property	Datatype	Single or repeating	Description
rest_store_type	string(1024)	S	<p>For the Azure Blob storage type, the value must be set to 0. For the Google Cloud storage type, the value must be set to 1.</p> <p> Note: Any value other than 1 for <i>rest_store_type</i> is considered for Azure Blob storage type only.</p> <p><i>OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS250400-AGD)</i> contains more information.</p>

Property	Datatype	Single or repeating	Description
store_params	string(1024)	R	<p>Configuration parameters for the Vault support. The configuration parameters are stored as key-value pair with the pre-defined keys.</p> <p><i>OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS250400-AGD)</i> contains more information.</p>

3.147 Retainer

Describes a retention policy.

3.147.1 Description

- Supertype: SysObject
- Subtypes: dmc_rps_retainer
- Internal name: dm_retainer
- Object type tag: 08

A retainer object records information about one retention policy. Retention policies are created through OpenText™ Documentum™ Content Management Retention Policy Services (accessed through Documentum Administrator). Retainer objects are owned by a member of the dm_retention_managers group or by the group itself. Retainer objects must be uniquely named within the repository and cannot be versioned.

3.147.2 Properties

The table describes the object properties.

Table 3-138: Retainer type properties

Property	Datatype	Single or repeating	Description
aging_method	integer	S	<p>Specifies the type of the retainer. Valid values are:</p> <ul style="list-style-type: none"> • 0: Fixed retention value • 1: Conditional (event-based) retention • 2: Retention hold
disposition_rule	integer	S	<p>Defines how OpenText Documentum Content Management (CM) Retention Policy Services disposes of an object. Valid values are:</p> <ul style="list-style-type: none"> • 0: DESTROY CHILDREN • 1: DESTROY_VDM • 2: DESTROY_ROOT

Property	Datatype	Single or repeating	Description
enforcement_rule	integer	S	<p>Defines how objects under this policy are managed when the retention period expires. Valid values are:</p> <ul style="list-style-type: none">• 0: Objects are retained until the retention rule is removed using Retention Policy Services.• 1: Documentum CM Server disallows deletion if the objects are linked to an active retainer.• 2: Objects are retained until their associated structural retainer or retainers are removed. <p>The retention policy disposition rules defined within Retention Policy Services control how objects associated with an active retainer are disposed.</p>

Property	Datatype	Single or repeating	Description
immutability_rule	integer	S	<p>Defines whether to mark documents or folders under the control of the policy immutable. Valid values are:</p> <ul style="list-style-type: none">• 0: Objects are not marked immutable• 1: Objects are marked immutable <p>The default is 0.</p> <p>When an object that is marked immutable due to its retention policy or policies is detached from all retainers, its <code>r_immutable_flag</code> property is not reset to F unless the object is the CURRENT version.</p>

Property	Datatype	Single or repeating	Description
r_retention_status	integer	S	<p>Specifies the status of the retainer. Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning active-objects under the control of the retainer may not be deleted and new objects may be added to the retainer • 1, meaning locked-the retainer is active but no new objects may be placed under its control • 2, meaning inactive-objects under the control of the retainer may be deleted from the repository <p>The default is 0.</p>
rendition_rule	integer	S	<p>Defines how multiple content files for one document are handled. Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning all content files associated with a document under the control of the policy are under retention • 1, meaning only the primary content files of a document under the control of the policy are under retention

Property	Datatype	Single or repeating	Description
retainer_root_id	ID	S	<p>Specifies the object to which this retainer is applied.</p> <p>If dm_retainer.retainer_strategy is set to 0 (object-based retention), retainer_root_id is the object ID of the individual object under retention.</p> <p>If retainer_strategy is set to 1 (container-based retention), retainer_root_id is the object ID of the folder containing the objects under retention.</p>
retainer_strategy	integer	S	<p>Defines whether the retainer controls a single object or multiple objects placed in a container. Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning the retainer controls a single object • 1, meaning the retainer controls multiple objects in a container <p>The default is 0.</p>
retention_date	date	S	Retention value when dm_retainer.retention_rule_type is set to 1, meaning date.
retention_interval	integer	S	<p>Retention value when dm_retainer.retention_rule_type is set to 2, meaning interval.</p> <p>The value is interpreted in seconds.</p>

Property	Datatype	Single or repeating	Description
retention_rule_type	integer	S	<p>Specifies how the retention period is defined. Valid values are:</p> <ul style="list-style-type: none">• 0, meaning there is no specified retention period• 1, meaning the retention period is defined as a specific date• 2, meaning the retention period is defined as an interval <p>The default is 0.</p>
retention_storage_classes	string(40)	S	Reserved for future use.
version_rule	integer	S	Defines how retention is applied to new versions of documents under the control of the retainer. The only valid value is 0, meaning that the retention period applies only to the specific version to which the retainer was attached.

3.148 Routecase condition

Records a route case condition expression for an automatic transition of a workflow activity.

3.148.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmc_routecase_condition
- Object type tag: 00

A routecase condition object records a conditional expression in a route case condition. You cannot create these objects directly. They are created when an addConditionRouteCase method (defined for the IDfActivity interface) is executed to save an activity's route case conditions when one or more of the route cases contains an XPath expression.

3.148.2 Properties

The table describes the object properties.

Table 3-139: Routecase condition type properties

Property	Datatype	Single or repeating	Description
a_attribute_name	string(32)	S	Name of the property referenced in the expression, if any.
a_boolean_value	Boolean	S	Value to be used in the comparison if the datatype identified in a_value_type is Boolean. This is not set if unless the datatype is Boolean.
a_double_value	double	S	Value to be used in the comparison if the datatype identified in a_value_type is double. This is not set if unless the datatype is double.

Property	Datatype	Single or repeating	Description
a_id_value	ID	S	<p>Value to be used in the comparison if the datatype identified in r_value_type is ID.</p> <p>This is not set if unless the datatype is ID.</p>
a_int_value	integer	S	<p>Value to be used in the comparison if the datatype identified in a_value_type is integer.</p> <p>This is not set if unless the datatype is integer.</p>
a_object_alias	string(128)	S	Name of the package, or manifest values referring to the workflow or work item.
a_relational_op	integer	S	<p>The relation operator in the condition. Valid values are:</p> <ul style="list-style-type: none"> • 0: = • 1: <> • 2: < • 3: > • 4: <= • 5: >=

Property	Datatype	Single or repeating	Description
a_repeating_attr_flag	integer	S	<p>Indicates whether the property named in a_attribute_name is a repeating property and if it is a repeating property, which values to examine when evaluating the condition. Valid values are:</p> <ul style="list-style-type: none"> • -1: The property is not a repeating property. • 0: ANY • 1: ALL • 2: FIRST • 3: LAST
a_string_value	string(1024)	S	<p>Value to be used in the comparison if the datatype identified in a_value_type is string.</p> <p>This is not set if unless the datatype is string.</p>
a_time_value	date	S	<p>Value to be used in the comparison if the datatype identified in a_value_type is Date.</p> <p>This is not set if unless the datatype is Date.</p>
a_value_type	integer	S	Data type of the value in the relational expression. The data type is expressed as an IDfValue constant.
a_xpath_datatype	string(64)	S	<p>The xschema's built-in datatype name.</p> <p>This is set only if the condition includes an XPath expression.</p>

Property	Datatype	Single or repeating	Description
a_xpath_def_namespace_uri	string(255)	S	The schema URI for the default XPath namespace. This is set only if the condition includes an XPath expression.
a_xpath_expression	string(1024)	S	An XPath expression. This is set only if the condition includes an XPath expression.
a_xpath_namespace_prefix	string(64)	R	The prefixes that alias the namespace URIs within the XPath expression. This is set only if the condition includes an XPath expression.
a_xpath_namespace_uri	string(255)	R	Namespace URIs of the XPath expression components. This is set only if the condition includes an XPath expression.
a_xpath_value	string(1024)	S	Literal value used in the XPath transition condition evaluation. The value is in the format in which it is found in the XML document. This is set only if the condition includes an XPath expression.
r_aspect_name	string(64)	R	Used internally.

3.149 S3 store

Represents a S3 content storage area.

3.149.1 Description

- Supertype: Store
- Subtypes: none
- Internal name: dm_s3_store
- Object type tag: inherited from Store

S3 store object represents a S3 content storage area. The properties in S3 store objects may be changed only by users with Sysadmin or Superuser user privileges.

3.149.2 Properties

The table describes the object properties.

Table 3-140: S3 store properties

Property	Datatype	Single or repeating	Description
store_params	string(1024)	R	<p>Configuration parameters for the proxy and WORM support. The configuration parameters are stored as key-value pair with the optional pre-defined keys.</p> <p><i>OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS250400-AGD)</i> contains more information.</p>

3.150 Scope config

Defines the context for the referenced display config objects.

3.150.1 Description

- Supertype: None
- Subtypes: None
- Internal name: dm_scope_config
- Object type tag: 6c

A scope config object defines the context for the referenced display config objects. Each display config object defines a set of properties and display hint for the properties. Each scope config object identifies one or more display config objects and one or more contexts in which the display config is used. Both scope config and display config objects are used by client applications. Documentum CM Server does not use these objects.

3.150.2 Properties

The table describes the object properties.

Table 3-141: Scope config type properties

Property	Datatype	Single or repeating	Description
display_config	ID	R	Object IDs of the display config objects used in the scopes defined in the scope config object.
scope_class	string(32)	R	The category of the scope. The only valid value is application.

Property	Datatype	Single or repeating	Description
scope_value	string(32)	R	<p>The specific scope within the scope class. For example, if scope_class is application, scope_value might be Webtop or a DAR file name.</p> <p>The scope at a particular index position is the specific instance of the class at the corresponding index position in scope_class.</p>

3.151 Script

Stores a script as content.

3.151.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_script
- Object type tag: 08

A script object stores a script as content. Script objects are used by OpenText Documentum CM clients. Their content files contain the scripts. Depending on where the script object is stored, the script is executed automatically when users connect to a repository or on demand, when users select the script from a menu. Users must have at least Read permission for the script object to execute its associated script.

3.151.2 Properties

The script type has no defined properties. It inherits all of its properties from its supertype, SysObject.

3.152 Sequence

Used by Documentum CM Server to generate object IDs.

3.152.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_sequence
- Object type tag: 20

A sequence object is used by Documentum CM Server to generate object IDs. There is one sequence object for each object type in a repository, up to a maximum of 255.

3.152.2 Properties

The table describes the object properties.

Table 3-142: Sequence type properties

Property	Datatype	Single or repeating	Description
i_current_mask	integer	S	Used internally to manage object IDs.  Note: When the object count exceeds four billion, capital letters and special characters are added to object IDs. In this case, the name of the object ID is called Galactic object ID.
i_high_mask	integer	S	Used internally to manage object IDs.
i_high_water_mark	integer	S	The highest number that can be assigned.

Property	Datatype	Single or repeating	Description
i_in_use	Boolean	S	Indicates if any of the numbers in the allowed range have been assigned.
i_last_no	integer	S	Contains the last number assigned.
i_low_mask	integer	S	Used internally to manage object IDs.
i_low_water_mark	integer	S	The first or lowest number that can be assigned.

3.153 Server config

Describes a Documentum CM Server.

3.153.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_server_config
- Object type tag: 3d

A server config object contains information that the server uses to define its operation and operating environment, such as the number of allowed concurrent sessions, maximum cache sizes, and the storage area locations, and the locations of executables that the server calls.

3.153.2 Properties

The table describes the object properties.

Table 3-143: Server config type properties

Property	Datatype	Single or repeating	Description
a_silent_login	Boolean	S	Used internally.
agent_launcher	string(32)	S	The name of the method object that launches the agent exec process. The default is agent_exec_method.

Property	Datatype	Single or repeating	Description
alias_set_id	ID	S	The object ID of the alias set object representing the system-level default alias set.
app_server_name	string(32)	R	Name of a Java servlet. The name <code>do_method</code> identifies the servlet to which <code>DO_METHOD</code> functions are directed.
app_server_uri	string(255)	R	URI for the servlet at the corresponding index position in <code>app_server_name</code> . Contains the host, port, and <code>servlet_path</code> in the following format: <code>http://host:port/ servlet_path</code> <i>host</i> is the IP address or name of the machine hosting the application server. <i>port</i> is the port number on which the application server is listening. <i>servlet_path</i> is the path to the servlet to which an <code>HTTP_POST</code> request is directed. The URI can contain only ASCII characters.

Property	Datatype	Single or repeating	Description
application_access_control	Boolean	S	T (TRUE) means users without Superuser privileges must use an application access token to connect to the server. F (FALSE) means any user can connect without an access token. The default is F.
assume_user_location	string(32)	S	Name of the location object for the directory containing the assume user program. The default is assume_user.
auth_plugin_location	string(32)	S	Name of the location object that identifies the directory location of authentication plug-ins. This is set to auth_plugin by default.
cached_types	string(32)	R	Names of the user-defined object types that you want to cache on server startup. The default is a single blank.
certdb_location	string(32)	S	Name of the location object pointing to the location of the certificate database. The default is ldapcertdb_loc. This property is deprecated.
change_password_location	string(32)	S	Name of the location object for the directory containing the change password program. The default is change_password.

Property	Datatype	Single or repeating	Description
checkpoint_interval	integer	S	Defines how often the server broadcasts its service information. The default is 300 (seconds).
client_cache_size	integer	S	Maximum allowed size of the client cache, expressed as the number of objects. The default is 50. The number of objects may exceed the specified maximum size if an object must be added to the cache and no candidate for deletion is found. Also, persistently cached objects are not counted towards the maximum.
common_location	string(32)	S	Name of the location object for the common directory. The default is common.
compound_integrity	Boolean	S	Indicates if the server is enforcing referential integrity for virtual documents. The default is TRUE.
concurrent_sessions	integer	S	Number of concurrent sessions. The default is 100.
dba_location	string(32)	S	Name of the location object that identifies the location of the dba directory in the OpenText Documentum CM installation. The default is dm_dba.

Property	Datatype	Single or repeating	Description
default_acl	integer	S	<p>Specifies which ACL the server uses as the default ACL when creating a new object if an ACL is not explicitly associated with the object. Valid values are:</p> <ul style="list-style-type: none"> • 1: ACL associated with the object's primary folder. • 2: ACL associated with the object's type. • 3: ACL associated with the user who created the object • 4: No default ACL is specified. <p>The default is 3.</p>
default_client_codepage	string(64)	S	<p>Specifies the default code page for clients. Valid values are:</p> <p>US-ASCII, UTF-8, ISO_8859-1, Shift_JIS, EUC-JP, EUC-KR, and ISO-10646-UCS-2.</p> <p>The value is determined programmatically and set during Documentum CM Server installation.</p>
events_location	string(32)	S	Name of the location object that identifies the events directory. The default is events.

Property	Datatype	Single or repeating	Description
extra_directory_config_id	ID	R	<p>Object IDs of the ldap config objects representing LDAP directories from which users and groups are synchronized.</p> <p>This property is deprecated.</p>
far_stores	string(32)	R	Names of the storage areas that are considered far for the server. The default is a single blank.
fulltext_location	string(64)	S	Name of the location object that identifies the location of the fulltext configuration file (dmfulltext.ini).
ignore_pre_processing	Boolean	S	Not currently used.
jms_config_id	string(32)	R	Object id of jms config objects from which current serer instance can be used.
jms_mode	integer	R	<p>Specifies to configure JMS high availability. Valid values are:</p> <ul style="list-style-type: none"> • 1: <i>Load balancing</i> • 2: <i>Failover</i> • 3: <i>Load balancing & Failover</i> <p> Note: At least one jms object should have either 1 or 3.</p>
jms_proximity	integer	S	Reserved to configure proximity setting.
jms_type	integer	S	Reserved to configure algorithm of load balancing for JMS. Currently JMS supports only Round Robin algorithm.

Property	Datatype	Single or repeating	Description
keep_entry_interval	integer	S	Defines how long a connection broker keeps a server entry in the absence of a checkpoint message from the server. The default is 1440 minutes (24 hours).
ldap_config_id	ID	S	<p>Object ID of the ldap config object representing the default LDAP directory used by Documentum CM Server.</p> <p>This property is deprecated.</p>
locale_name	string(32)	S	<p>Documentum CM Server's locale. Legal values are:</p> <ul style="list-style-type: none"> • en, for English • es, for Spanish • fr, for French • de, for German • it, for Italian • sv, for Swedish • ja, for Japanese • ko, for Korean • zh, for Chinese <p>The value is determined programmatically and set during Documentum CM Server installation.</p>
log_location	string(32)	S	Name of the location object for the logs directory. The default is log.

Property	Datatype	Single or repeating	Description
login_ticket_timeout	integer	S	Defines the length of time for which a login ticket is valid. The value is interpreted in minutes. The minimum accepted value is 1 minute. The default is 5 minutes.
mail_method	string(32)	S	<p>Name of the method that Documentum CM Server uses to send email messages.</p> <p>Valid values are dm_event_sender, dm_event_sender_java, an empty string, or dm_event_template_sender. The default value is dm_event_sender_java.</p> <p>dm_event_sender_template is a valid value only if you have Business Process Designer installed and are using the workflow email template feature.</p> <p>Setting this property to an empty string does not disable server email. If you want to disable server email, you must set the mail_notification key in the server.ini file.</p>

Property	Datatype	Single or repeating	Description
max_cm_parallel_degree	integer	S	Defines maximum sessions created for content migration. The default value is 10, but can be modified by setting the max_cm_parallel_degree parameter in dm_server_config. The valid range for this parameter is 2 to 50.
max_login_ticket_timeout	integer	S	Maximum length of time, in minutes, that a login ticket can remain valid. The minimum value is 1 minute. The default value is 43200 minutes (30 days).
nfs_enabled	Boolean	S	Indicates if the server is using NFS for file sharing. The default is FALSE.
nls_location	string(32)	S	Name of the location object for the nls directory. The default is a single blank.

Property	Datatype	Single or repeating	Description
object_name	string(32)	S	<p>Specifies the server config object. The default is the name of the repository for which the server was started.</p> <ul style="list-style-type: none"> • Server config names for a Documentum CM Server at a primary site is <=32 characters in length. • Server config names for content-file servers (remote Content Servers) are 25 characters or less. • Server config names are created automatically when the server is installed.
operator_name	string(255)	S	Specifies the operator for the repository (for archiving and restore operations). The default is the value in the server config object's owner_name property.
projection_enable	Boolean	R	Indicates whether projection to the connection broker specified at the corresponding index position in projection_targets is enabled.
projection_netloc_enable	Boolean	R	Indicates whether projection to the network location specified at the corresponding index position in projection_netloc_id is enabled.

Property	Datatype	Single or repeating	Description
projection_netloc_id	string(80)	R	User-defined identifiers of network locations.
projection_notes	string(80)	R	User-defined.
projection_ports	integer	R	Specifies the port on which the connection broker is listening. The value at each index level is matched to the connection broker specified at the corresponding level in projection_targets. The default is 1489.
projection_proxval	integer	R	<p>Proximity values projected to the connection brokers, the network locations, or both.</p> <p>The value at each index position is projected to the connection broker specified at the corresponding position in projection_targets if projection_enable is TRUE in the corresponding index position.</p> <p>Similarly, the value at each index position is projected to the network location specified at the corresponding index position in projection_netloc_id if projection_netloc_enable is TRUE in the corresponding index position.</p>

Property	Datatype	Single or repeating	Description
projection_targets	string(80)	R	Names of the host machines on which the connection brokers reside.
r_castore_enabled	Boolean	S	Specifies whether the server was installed with a Content Services for Centera. Valid values are: <ul style="list-style-type: none"> • 1, meaning the server was installed with a Content Services for Centera, default value is 1. • 2, meaning the server was not installed with a Centera.
r_host_name	string(64)	S	Name of the host machine on which the server resides.
r_install_domain	string(16)	S	Name of the Windows domain.
r_install_owner	string(255)	S	Name of the installation owner.
r_process_id	integer	S	Contains the process id of the main server thread (parent server).
r_server_version	string(32)	S	Version number of the server.
r_trusted_mode	integer	S	Specifies whether the server is a trusted server. 0 means that the server is not a trusted server. 1 means the server is a trusted server. Default value is 1.
rend_backing_store	string(32)	S	Obsolete.

Property	Datatype	Single or repeating	Description
restrict_su_ticket_login	Boolean	S	<p>T (TRUE) means a user with Superuser privileges cannot connect to the server using a global login ticket.</p> <p>F (FALSE) allows users with Superuser privileges to connect to the server using a global login ticket.</p> <p>The default is F.</p>
rightsight_image	string(255)	S	<p>Used to construct the URL when a user is sending a document to a mail recipient. Valid values are:</p> <ul style="list-style-type: none"> • webtop • a RightSite image <p>RightSite images are found in the Windows registry. An example of a Rightsite image is /rs-bin/RightSite.dll</p> <p>The default is webtop.</p>
secure_connect_mode	string(16)	S	<p>The type of port on which the server is listening. Valid values are:</p> <ul style="list-style-type: none"> • native, meaning an unsecure port • secure, meaning a secure port (using SSL) • dual, meaning both a native and a secure port <p>Non-trusted servers do not use this property. They accept only native connections.</p>

Property	Datatype	Single or repeating	Description
secure_writer_location	string(32)	S	Name of the location object for the directory containing the secure writer program. The default is secure_common_area_writer.
server_cache_size	integer	S	Maximum allowed size, expressed as the number of objects. The default is 200.
server_os_codepage	string(64)	S	The code page used by the operating system of the Documentum CM Server host. Legal values vary by operating system. Valid options are: US-ASCII UTF-8 ISO_8859-1 Shift_JIS EUC-JP EUC-KR ISO-10646-UCS-2 The value is determined programmatically and set during Documentum CM Server installation.
sibling_checkpoint_interval	integer	S	Defines how often the server broadcasts sibling information (Not currently used).
sibling_export_enabled	Boolean	S	Indicates if the server can broadcast information about its sibling servers. (Not currently used).
signature_chk_loc	string(32)	S	Name of the location object for the directory containing the signature validation program. The default is validate_signature.

Property	Datatype	Single or repeating	Description
smtp_server	string(64)	S	Name of the SMTP server host. This is used for email notification system. This is mandatory for Windows and optional for Linux.
stage_destroyer_location	string(32)	S	Not currently used.
system_converter_location	string(255)	S	Name of the location object for the directory containing the convert.tbl file and the system-supplied transformation scripts. The default is convert.

Property	Datatype	Single or repeating	Description
system_shutdown_timeout	integer	S	<p>Specifies the time in seconds that the workflow agent attempts to shut down work items gracefully after receiving a shutdown command.</p> <p>If the shutdown timeout period is exceeded (or is set to zero), Documentum CM Server takes over and shuts down the workflow agent immediately.</p> <p>If the shutdown timeout period is a negative value, Documentum CM Server waits for the workflow agent to complete all automated tasks before shutting down gracefully.</p> <p>The default value is 120 seconds. Use Documentum Administrator to change the setting.</p>
temp_location	string(32)	S	Name of the location object for the temp directory. The default is temp.
turbo_backing_store	string(32)	S	Contains the name of the file store object for the file store storage area where the server puts renditions generated by indexing blob and turbo content. The default is filestore_01.

Property	Datatype	Single or repeating	Description
user_converter_location	string(32)	S	Name of the location object that identifies the directory containing the user-defined transformation scripts. The default is a single blank.
user_validation_location	string(32)	S	Name of the location object for the directory containing the external password checking program. The default value is validate_user.
verit_location	string(32)	S	Obsolete.
web_server_loc	string(255)	S	Name of the Web server's host machine, and optionally, its domain and protocol. Examples: cinderella cinderella.mycompany.com http://cinderella.mycompany.com The default protocol is http.
web_server_port	integer	S	Specifies the port which the Web server is using. The default value is 80.

Property	Datatype	Single or repeating	Description
wf_agent_notify_new_task	Boolean	S	<p>Forces immediate processing of automatic activities by notifying the workflow agent to process automatic tasks immediately after the system creates them.</p> <p>By default, the value is TRUE.</p> <p>When this attribute is set to TRUE, the following occurs:</p> <ul style="list-style-type: none"> • If the system creates automatic tasks when the workflow agent is processing the previously collected tasks, then the workflow agent ignores the wf_sleep_interval and collects the next set of automatic activities. • If the system has not created an automatic task when the workflow agent is processing the previously collected tasks, then the workflow agent sleeps for the time specified in wf_sleep_interval. • If the system creates tasks when the workflow agent is sleeping, then the agent wakes up immediately and collects the next

Property	Datatype	Single or repeating	Description
			<p>set of automatic activities.</p> <p>When this attribute is set to FALSE, the workflow agent honors the value of the wf_sleep_interval and sleeps for the time specified before the workflow agent collects the next set of automatic activities for processing.</p>
wf_agent_worker_threads	integer	S	<p>Number of workflow agent worker sessions. The maximum value is 1000. The default value is 3. Setting this to 0 disables the workflow agent.</p>
wf_sleep_interval	integer	S	<p>Length of time, in seconds, that the master workflow agent master session waits before querying the repository for activities awaiting execution in the absence of a notification of an activity waiting for execution.</p> <p>The default is 5 seconds. There is no maximum value.</p>

3.154 Session

Not currently used.

3.154.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_session
- Object type tag: 01

The session object type is constructed by the server during server start-up. It is not currently used by Documentum CM Server, and there are no objects of that type stored in the repository. However, you can execute a describe command through IDQL to view its properties, and it is visible as a dm_type object.

3.155 Smart list

Represents the set of objects determined by the SELECT query defined for the Smart List.

3.155.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_smart_list
- Object type tag: 08

A Smart List object represents the set of objects determined by the SELECT query defined for the Smart List. When a user opens a Smart List, the query is executed and the retrieved items are displayed to the user in a window that allows the users to manipulate the objects.

A Smart List object can only be created using Desktop Client or Webtop. On Webtop, the Smart List object contains the saved search object created by the search service. This object contains the search definition and search results. The Smart List object created by Desktop Client can be executed in Webtop but not edited nor listed as a saved search.

If the Smart List is created from Desktop Client, the query is saved as DQL. If the Smart List is created from Webtop, the query is saved as an XML file. Smart Lists created in Desktop Client can be read by Webtop. However, Smart Lists created in Webtop cannot be read by Desktop Client.

3.155.2 Properties

The table describes the object properties.

Table 3-144: Smart list type properties

Property	Datatype	Single or repeating	Description
has_results	Boolean	S	TRUE if the saved search was saved with results.
query_type	string(32)	S	Specifies the search type. Valid values are: <ul style="list-style-type: none"> • passthrough_query • query_builder • blank when query type is unknown
results_count	integer	S	Number of results the search had when the search was saved.
selected_sources	string(256)	R	List of the sources searched by the search. These can be repositories or external sources.

3.156 SMTP config

Specifies the properties required for the SMTP server.

3.156.1 Description

- Supertype: None
- Subtypes: None
- Internal name: dm_smtp_config
- Object type tag: 00

SMTP config object stores the SMTP server parameters required for mail notification including custom port, SSL, and authentication. This configuration works only when the mail_method configured in dm_server_config is dm_event_sender_java. A repository can have only one SMTP config object. Only superusers have the create, update, view, and delete permissions. To change or modify the settings, use IAPI/IDQL or Foundation Java API. Any modification to SMTP config object requires JMS restart.

3.156.2 Properties

The table describes the object properties.

Table 3-145: SMTP config type properties

Property	Datatype	Single or repeating	Description
smtp_host	string(1024)	S	Host name or IP address of the SMTP server.
smtp_port	integer	S	Port reserved for the SMTP server. The default port is 25 for native and 465 for secure connections.
auth_enable	boolean	S	Specifies if SMTP server requires authentication. The default value is false.
smtp_user	string(255)	S	<ul style="list-style-type: none">• User name to access the SMTP server.• Client ID to access the Microsoft Copilot 365 server.
smtp_pass	string(255)	S	<ul style="list-style-type: none">• Password of the user to access the SMTP server.• Client secret to access the Microsoft Copilot 365 server.
ssl_enable	boolean	S	Specifies if SMTP server requires SSL connection. The default value is false.
smtp_params	string(1024)	R	Tenant ID, Proxy Host, and Proxy Port used to access the Microsoft Copilot 365 server.

3.157 SSA policy

Defines a content assignment policy for new content files.

3.157.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_ssa_policy
- Object type tag: 08

An ssa policy object defines storage rules for content files. The rules are stored as content of the object. A policy is associated with one or more object types through a relationship, recorded in an relation ssa policy object.

SSA policy objects are created using Documentum Administrator. Existing policies can be copied or modified, but not versioned.

3.157.2 Properties

The table describes the object properties.

Table 3-146: SSA policy type properties

Property	Datatype	Single or repeating	Description
is_activated	Boolean	S	<p>Specifies whether the policy is active (being enforced by the policy engine). T (TRUE) means the policy is enforced. F (FALSE) means the policy is inactive.</p> <p>The default is T.</p>

3.158 State extension

Specifies the lifecycle state with which a state extension object is associated.

3.158.1 Description

- Supertype: Relation
- Subtypes: Retention Policy Services Phase Rel, Retention Policy Services Retainer Event Rel
- Internal name: dm_state_extension
- Object type tag: 37

A state extension object identifies the lifecycle state with which the state extension object is associated. The state extension object type is the base object type for state extension object subtypes. Typically, the state extension object type is subtyped to add properties to store the application-specific information needed for particular lifecycle states.

3.158.2 Properties

The table describes the object properties.

Table 3-147: State extension type properties

Property	Datatype	Single or repeating	Description
state_no	integer	S	Number of the state with which the state extension object is associated. The number is the value in the state_no property of the dm_policy object.

3.159 State type

Specifies a particular lifecycle state type name and application pair.

3.159.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dm_state_type
- Object type tag: 00

A state type object identifies a particular lifecycle state type name and application pair. State type objects can be used to associate a state type name with a particular application. Some OpenText Documentum CM client applications require specific state type names be assigned to lifecycle states. User applications can use state types.

3.159.2 Properties

The table describes the object type properties.

Table 3-148: State type properties

Property	Datatype	Single or repeating	Description
application_code	string(32)	S	Name of the application that recognizes the state type.
state_type_name	string(32)	S	Name of a lifecycle state type.

3.160 Store

Represents a content storage area.

3.160.1 Description

- Supertype: Persistent Object
- Subtypes: Distributed Store, File Store, Linked Store, Blob Store, External Store, CA Store, Atmos Store, ViPR store, S3 store
- Internal name: dm_store
- Object type tag: 0e

A store object represents a content storage area. The properties in store objects may be changed only by users with Sysadmin or Superuser user privileges.

3.160.2 Properties

The table describes the object properties.

Table 3-149: Store type properties

Property	Datatype	Single or repeating	Description
base_url	string(255)	S	The basic URL used to retrieve contents directly from a storage area.
capacity	integer	S	Not currently used.
component	string(64)	R	The name of the storage objects representing component storage areas. Used by linked store and distributed stores.
compression_mode	integer	S	<p>Applicable only to file store and ca store storage areas, this property indicates whether content written to the storage area is compressed. Valid values are:</p> <ul style="list-style-type: none">• 0, meaning the content in the storage area is not compressed• 1, meaning the content is compressed <p>If not explicitly set when the storage area is created, the default value is 0.</p> <p>The setting may not be changed after the storage area is created.</p>

Property	Datatype	Single or repeating	Description
content_dupl_pref	integer	S	<p>This property indicates the content duplication preference for the storage area. Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning Documentum CM Server does not check the storage area for duplicate content files when saving content files to the storage area • 1, meaning Documentum CM Server checks the storage area for duplicate content files when saving content files to the storage area, and uses a SHA1 hash • 3, meaning Documentum CM Server checks the storage area for duplicate content files when saving content files to the storage area, and uses a SHA2 (256) hash • 4, indicates Documentum CM Server checks the storage area for duplicate content files while saving content files to the storage area, and uses a SHA3-384 hash. <p> Note: You must use 4 for release version 20.2 and later.</p>

Property	Datatype	Single or repeating	Description
			<p>The default is 0.</p> <p>For more information about how this property is used, see <i>OpenText Documentum Content Management - Administrator User Guide</i> (EDCAC250400-UGD).</p>

Property	Datatype	Single or repeating	Description
content_hash_mode	integer	S	<p>This property configures hash generation for content files in the storage area. This property setting is applicable only to file store storage areas.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning do not generate hashes for the content files in the storage area • 1, meaning generate a SHA1 hash for each content file in the storage area • 3, meaning generate a SHA2 (256) hash for each content file in the storage area • 4, indicates generate a SHA3-384 hash for each content file in the storage area. <p> Note: You must use 4 for release version 20.2 and later.</p> <p>If content_dupl_pref is set to a non-zero value, the value of content_hash_mode must be 1 or 3. Documentum CM Server sets the value automatically under that condition.</p> <p>If content_dupl_pref is 0, the default value for</p>

Property	Datatype	Single or repeating	Description
			<p>content_hash_mode is 0.</p> <p>For more information about the content_hash_mode property, see <i>OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS250400-AGD)</i>.</p>
credential_id	string(255)	S	The ID required for access to the store.
credential_key	string(255)	S	The encrypted key or password for the given credential_id to access the store.
crypto_key	string(256)	S	<p>For a local key management repository, contains the symmetric key used to encrypt the content files stored in the storage area.</p> <p>For a remote key management repository, contains the ID of the symmetric key used to encrypt the content files stored in the storage area.</p> <p>You cannot create indexes on this property.</p>

Property	Datatype	Single or repeating	Description
crypto_mode	integer	S	<p>Specifies whether the contents of the storage area are encrypted, and what encryption algorithm is used. Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning the contents are not encrypted • 1, meaning the contents are encrypted <p>This value is set when the storage area is created and cannot be changed afterwards. You can only set this value for file store storage areas.</p> <p>The default value is 0.</p>
current_use	integer	S	For internal use.
default_retention_days	integer	S	<p>Number of days to retain content in the storage area. The count of days begins when the content is saved to the storage area.</p> <p>This property is used only for ca store (retention-enabled) storage areas.</p> <p>If this property is set, its value takes precedence over a value set in dm_ca_store.a_default_retention_date.</p>

Property	Datatype	Single or repeating	Description
digital_shredding	Boolean	S	<p>Specifies whether digital shredding is enabled for the storage area. Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning shredding is disabled • 1, meaning shredding is enabled <p>The default value is 0.</p> <p>This property only affects file store storage areas.</p>
get_method	string(32)	S	Contains the name of the dm_method object representing the user-defined program that retrieves a specified content.
i_use_mask_after	integer	S	Used internally to manage storage ticket numbers for content files.
java_access	string(255)	S	A name or identifier of the plug-in module implementing access to the store. For a dm_atmos_store this is expected to be the name of a dmc_module.

Property	Datatype	Single or repeating	Description
media_type	integer	S	<p>Specifies the media type of content stored in this storage area. Valid values are:</p> <ul style="list-style-type: none"> • 0, for any type other than thumbnail or streaming • 1, for thumbnail • 2, for streaming <p>The property cannot be changed after the storage area is created.</p>
name	string(64)	S	Contains the name of the store object. This must be a unique name within the repository.
native_access	string(255)	S	A value of none indicates that the Documentum CM Server does not have direct access to the store and has to rely on a separate process such as Accelerated Content Services. An empty value is not equivalent to a value of none. An empty value indicates that Documentum CM Server will continue to use existing internal implementations to access these stores.
offline_get_method	Boolean	S	Indicates whether the application should regard the retrieved content as immediately available or awaiting restoration (from archiving).

Property	Datatype	Single or repeating	Description
r_component	ID	R	The object IDs of any component storage areas.
r_component_count	integer	S	The total number of component storage areas.
r_status	integer	S	Records the state of the storage area. Valid values are: <ul style="list-style-type: none">• 0, for on-line• 1, for off-line• 2, for read-only
require_ticket	Boolean	S	Indicates whether Documentum CM Server generates a ticket when returning a URL to a content file. TRUE, meaning a ticket is returned. FALSE meaning a ticket is not returned. The default is TRUE.

Property	Datatype	Single or repeating	Description
store_type	integer	S	<p>Value indicating the subtype of the storage area. This can be one of the following values:</p> <ul style="list-style-type: none"> • 1, for file store • 3, for linked store • 4, for distributed store • 5, for blob store • 7, for external store • 8, for external file store • 9, for external URL store • 10, for external free store • 11, for CA store <p> Note: 2 and 6 are currently unused values.</p>

3.161 Subcontent

Stores a content file in turbo storage or the content addresses of content stored in content-addressed storage systems.

3.161.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_subcontent
- Object type tag: 64

A subcontent object stores a content file in turbo storage or the content addresses of content stored in content-addressed storage systems. When a content file in turbo storage is too big to store in a content object, the content is stored in subcontent object. If a content file stored in a content-addressed system is modified or if its metadata properties in the storage system are modified, the storage system creates a new address for the content file. This address is stored in a subcontent object.

Subcontent objects are internal objects that users and applications cannot access directly. However, users and applications can query subcontent objects using DQL.

3.161.2 Properties

The table describes the object properties.

Table 3-150: Subcontent type properties

Property	Datatype	Single or repeating	Description
parent_id	string(16)	S	Object ID of the dmr_content object that represents the content file or content address stored in this object.
is_other	Boolean	S	Is true when this object specifies resource fork content. Not implemented yet in 6.6 release.
i_contents	string() The size of the string is the maximum for the underlying database.	R	For objects in turbo storage, this contains the object's content file if the content is too big for i_contents in the content object. This property cannot be selected using either the API or DQL when it used as a storage area for turbo content. For objects in content-addressed storage, this contains additional content addresses generated by operations on the object or its storage metadata values. All addresses point to the same content, however the metadata and retention period information associated with a particular address typically differs.

Property	Datatype	Single or repeating	Description
i_contents_size	integer	R	<p>Actual size of the content, including trailing spaces, stored in the corresponding index position of i_contents.</p> <p>This property is not used if the subcontent object records a content address.</p>
i_partition	integer	S	Specifies the partition where the object resides.

3.162 SysObject

Serves as the parent type of the most commonly used objects in the OpenText Documentum CM system.

3.162.1 Description

- Supertype: Persistent Object
- Subtypes: Document, Folder, Output Device, Category Class, and others, too many to list
- Internal name: dm_sysobject
- Object type tag: 08

The SysObject type is the parent type of the most commonly used objects in the OpenText Documentum CM system. The SysObject type has three important properties, represented by properties, that it passes on to all its subtypes. These properties are:

- A SysObject accepts security permissions. Properties defined for the SysObject allow you to set permissions on the object.
- A SysObject, unless it is a cabinet, can belong to a folder.
- A SysObject can own one or more content objects.

3.162.2 General properties

This table describes the general object properties.

Table 3-151: General SysObject type properties

Property	Datatype	Single or repeating	Description
a_application_type	string(32)	S	Reserved for use by OpenText Documentum CM products.
a_archive	Boolean	S	Used internally.
a_extended_properties	string(32)	R	Values of the relation_name property in a dm_relation object or subtype of a dm_relation. This property is used with the r_object_id of the sysobject to identify the relation object that contains the extended properties of the SysObject .
a_is_hidden	Boolean	S	Indicates if this object is visible to end-users.
a_is_template	Boolean	S	Indicates whether the object is a template.
a_last_review_date	Date	S	Date of the last review of the object. Some OpenText Documentum CM client products set this value automatically.
a_retention_date	date	S	Used internally for Retention Policy Services. Records the (computed) disposition date for objects under retention.

Property	Datatype	Single or repeating	Description
a_special_app	string(32)	S	Reserved for use by OpenText Documentum CM products.
a_status	string(16)	S	Reserved for use by OpenText Documentum CM products.
authors	string(48)	R	List of the authors for the object. This property is user-defined.
i_is_deleted	Boolean	S	If the object is the root version of a version tree, deleting the object sets this property to TRUE.
i_is_reference	Boolean	S	Indicates whether the object is a mirror object for a reference link to a remote object. TRUE means that the object is a mirror object.
i_contents_size	integer	R	Actual size of the content, including trailing spaces, stored in the corresponding index position of i_contents. This property is not used if the subcontent object records a content address.
i_partition	integer	S	Specifies the partition where the object resides.

Property	Datatype	Single or repeating	Description
i_property_bag	string(2000)	S	The property stores the names and values of non-qualifiable properties. It is also used to store the names and values of aspect properties if the properties are added to the aspect with the OPTIMIZEFETCH option.
i_retainer_id	ID	R	Object IDs of the retainer representing the retention policy that controls the object.
keywords	string(48)	R	List of user-defined keywords for the object.
language_code	string(5)	S	Five-character code, in the format <i>xx_yy</i> , indicating the language in which the document is written and the country of origin. The first two characters (<i>xx</i>) contain the language code and the final two (<i>yy</i>), the country code.
object_name	string(255)	S	Name of the object. This property is user-defined. If you are using a OpenText Documentum CM client, this value is the name that appears under the icon of the object.
r_access_date	date	S	Contains the date and time when a Getfile or Print method last accessed this object.

Property	Datatype	Single or repeating	Description
r_alias_set_id	ID	S	Object ID of the alias set associated with the sysobject.
r_aspect_name	string(64)	R	Reserved for internal use. Manipulation of this property by users or external applications is not supported.
r_creation_date	date	S	Date the object was created. The server sets this value.
r_modifier	string(255)	S	Name of the user who made the last modification.
r_modify_date	time	S	Date the object was last modified. The server sets this value.
r_object_type	string(32)	S	The type of the object, for example, dm_sysobject or dm_document, set when the object is created.
r_property_bag	string(2000)	R	This property stores overflow from the i_property_bag property.
subject	string(192)	S	Subject of the object. This is user-defined.
title	string(400)	S	Title of the object. This is user-defined.

3.162.3 Folder properties

This table describes the object properties that apply to folders.

Table 3-152: SysObject folder properties

Property	Datatype	Single or repeating	Description
i_folder_id	ID	R	Object IDs of all folders linked to the object.

Property	Datatype	Single or repeating	Description
i_has_folder	Boolean	S	Flag indicating whether this object is the CURRENT object in the version tree. T means the object has the CURRENT version label. F means it does not.
r_link_cnt	integer	S	Number of objects linked to the folder. This property is used for folders and virtual documents.
r_order_no	integer	R	No longer used. Preserved for backwards compatibility.
i_reference_cnt	integer	S	Number of folder references made to this object.
r_component_label	string(32)	R	No longer used. Preserved for backwards compatibility.
r_composite_id	ID	R	No longer used. Preserved for backwards compatibility.
r_composite_label	string(32)	R	No longer used. Preserved for backwards compatibility.

3.162.4 Virtual document properties

This table describes the SysObject properties that apply the virtual documents.

Table 3-153: Virtual document-related properties

Property	Datatype	Single or repeating	Description
a_compound_architecture	string(16)	S	Used by Virtual Document Manager to determine if the object can be structurally changed through VDM. An empty string indicates that the object can be changed.
a_link_resolved	Boolean	S	Used internally.
resolution_label	string(24)	S	Contains the default version label used to resolve late-bound nodes of a virtual document.
r_assembled_from_id	ID	S	Object ID of the virtual document that was the source of the assembly associated with a document.
r_frn_assembly_cnt	integer	S	Contains a count of the number of frozen assemblies that contain this object.
r_has_frn_assembly	Boolean	S	Indicates that the document's assembly is frozen.
r_is_virtual_doc	integer	S	Indicates whether the SysObject is a virtual document. If the value is 1, the object is a virtual document. If the value is 0, the object is not a virtual document unless the r_link_cnt is greater than 0.
r_link_cnt	integer	S	Number of components in the virtual document.

Property	Datatype	Single or repeating	Description
r_link_high_cnt	integer	S	<p>Records the current maximum order number assigned to a component.</p> <p> Note: Enable the DM_UPDATE_VDM_HIGH_LINK_COUNT environment variable and set it to 1. This allows server to update this property, which Foundation Java API uses to calculate the correct order number when more than 7 nodes are to be inserted in a virtual document.</p>

3.162.5 Content properties

This table describes the content properties

Table 3-154: SysObject content properties

Property	Datatype	Single or repeating	Description
a_content_type	string(32)	S	<p>File format of the content of the object.</p> <p>For some system-created renditions, Documentum CM Server sets this value to dm_internal.</p>
a_full_text	Boolean	S	Indicates whether the document is marked for full-text indexing. The default is TRUE.

Property	Datatype	Single or repeating	Description
a_is_signed	Boolean	S	Indicates whether the object has been digitally signed. The property is set to T by the client application after the user signs the object.
a_storage_type	string(64)	S	Specifies the storage area for content files associated with the object. This is the name of the storage object representing the storage area. The size of this property was changed from 32 to 64 in release 6.6.
i_contents_id	ID	S	Object ID of the content object for an object that has only one content. This is not used if an object has multiple content. In such cases, the Content Facility manages the relationships between the object and its multiple content.
r_content_size	integer	S	Size, in bytes, of the first content file associated with the document. This property cannot record content sizes greater than 2 GB. Examine r_full_content_size to obtain the size of content larger than 2 GB.

Property	Datatype	Single or repeating	Description
i_retain_until	date	S	Most restrictive retention date among all content associated with this object. This property is only set if the content has a retention period. Retention periods are set by a retention policy or through storage in a retention-enabled storage area with a retention period.
r_full_content_size	double	S	Size, in bytes, of the first content file added to the SysObject.
r_page_cnt	integer	S	Number of content files associated with the object.

3.162.6 Web properties

The Web properties described in this table are used by the OpenText Documentum CM applications for managing SysObjects that are published on Web sites. These properties are not intended for user applications.

Table 3-155: SysObject Web properties

Property	Datatype	Single or repeating	Description
a_category	string(64)	S	Used internally to manage the application.
a_effective_date	date	R	The date on which the document can be published to the Web site. The value at a particular index position applies to the Web site named at the corresponding index position in a_effective_label.

Property	Datatype	Single or repeating	Description
a_effective_flag	string(8)	R	Indicates whether a pending expiration notice was sent to the document's owner. The value at a particular index position applies to the Web site named at the corresponding index position in a_effective_label.
a_effective_label	string(32)	R	<p>User-defined, typically a symbolic label.</p> <p>The WebCache export operation examines the values in the a_effective_date and a_expiration_date properties of all documents whose a_effective_label property value matches the webc config effective_label property.</p> <p>If unspecified, effective labels are not enforced.</p>
a_expiration_date	date	R	The date at which the document is to be removed from the Web site. The value at a particular index position applies to the Web site named at the corresponding index position in a_effective_label.

Property	Datatype	Single or repeating	Description
a_publish_formats	string(32)	R	Lists the object's renditions that are to be exported to the WebCache repository. The system will export these renditions plus those defined in the source_formats property of the webc config object. This is an optional property. If unspecified, only renditions defined in the webc config object are published.

3.162.7 Security properties

The security properties described in this table provide information that allows the server to enforce security on the object. Changes to security-related properties can create a new ACL for the object.

Table 3-156: SysObject security properties

Property	Datatype	Single or repeating	Description
a_controlling_app	string(32)	S	Specifies the application or applications that can modify this object. If NULL, then any application can modify the object.
acl_domain	string(255)	S	Specifies the domain of the ACL associated with the object. The value will be either the user who created the ACL or, for system-level ACLs, the name of the repository owner.
acl_name	string(32)	S	The object name of the associated ACL.
group_name	string(255)	S	Group to which this object belongs.

Property	Datatype	Single or repeating	Description
group_permit	integer	S	Object-level permission assigned to the object's group for this object.
i_cabinet_id	ID	S	Object ID of the cabinet that is the object's primary storage location.
owner_name	string(255)	S	Name of the object's owner. This can be a user or group name.
owner_permit	integer	S	Object-level permission assigned to the owner for this object.
r_creator_name	string(255)	S	Name of the creator of the object.
r_is_public	Boolean	S	<p>Value is computed based on the ACL of the object.</p> <p>TRUE means that dm_world has at least Read permission. FALSE indicates that dm_world does not have Read or higher permission.</p>
r_lock_machine	string(80)	S	<p>Name of the client machine on which a user is working when he or she locks an object (by a checkout or branch operation).</p> <p>If the server is unable to resolve the name, r_lock_attribute is set to the client machine's IP address instead.</p>
world_permit	integer	S	Object-level permission assigned to the world (all users except owner and group members) for this object.

3.162.8 Version properties

The version properties described in this table provide information that lets the server manage and track the versions of an object.

Table 3-157: SysObject version properties

Property	Datatype	Single or repeating	Description
i_antecedent_id	ID	S	Object ID of the parent version of the object.
i_branch_cnt	integer	S	Number of branches on the version tree that contains the object.
i_chronicle_id	ID	S	Object ID of the root object of the version tree that contains this object.
i_direct_dsc	Boolean	S	Indicates whether the object has any direct descendants (versions derived directly from this object). The default is FALSE.
i_latest_flag	Boolean	S	Indicates whether this version is the most recent version of the object on a particular branch in the version tree.
log_entry	string(120)	S	Comments specified by the user.
r_frozen_flag	Boolean	S	Indicates whether the object is unchangeable because it was specifically frozen. The default is FALSE.
r_immutable_flag	Boolean	S	Indicates whether the object can be changed. The default is FALSE.
r_lock_date	date	S	Date that this object was locked.

Property	Datatype	Single or repeating	Description
r_lock_owner	string(255)	S	Name of the user who locked the object.
r_version_label	string(32)	R	List of the version labels associated with the object. The first position in this property holds the object's implicit version label. The remaining rows contain the object's symbolic version labels.

3.162.9 Event properties

This table describes the SysObject properties that apply to events.

Table 3-158: SysObject event properties

Property	Datatype	Single or repeating	Description
r_has_events	Boolean	S	Indicates whether any users have registered to receive events for this object. The default is FALSE.

3.162.10 Lifecycle properties

This table describes the SysObject properties that apply to lifecycles (policy objects).

Table 3-159: SysObject lifecycle properties

Property	Datatype	Single or repeating	Description
r_policy_id	ID	S	Object ID of the associated policy object.
r_current_state	integer	S	State number of the current state of the object in the lifecycle.

Property	Datatype	Single or repeating	Description
r_resume_state	integer	S	The state number to which the object is resumed if r_current_state identifies an exception state.

3.163 Sysprocess config

Reserved for internal use.

3.163.1 Description

- Supertype: SysObject
- Subtypes: JMS Config
- Internal name: dm_sysprocess_config
- Object type tag: 08

Do not use this type. It is reserved for internal use only.

3.164 Taxonomy

Records the default values for the properties in the categories of a taxonomy structure in the repository.

3.164.1 Description

- Supertype: Category
- Subtypes: None
- Internal name: dm_taxonomy
- Object type tag: 0b

A taxonomy object records the default values for the properties in the categories of a taxonomy structure in the repository.

3.164.2 Properties

The table describes the object properties.

Table 3-160: Taxonomy type properties

Property	Datatype	Single or repeating	Description
taxonomy_version	string(32)	S	Current version of the installed taxonomy.

3.165 TCF activity template

Represents one lifecycle state action.

3.165.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dmc_tcf_activity_template
- Object type tag: 09

A tcf activity template object represents one action that may be executed when an object enters a lifecycle state. A suite of TCF activity template objects is installed with the Lifecycle Editor. The content files associated with the objects are XML files that contain the template's string identifier and the parameters that must be supplied to execute the actions. You cannot create these objects manually.

3.165.2 Properties

The table describes the object properties.

Table 3-161: TCF activity template type properties

Property	Datatype	Single or repeating	Description
act_template_identifier	string(256)	S	Character string identifier that identifies this tcf activity template object.
template_groups	string(32)	S	Reserved for future use.

3.166 TCF activity

Records a sequence of actions to be executed on entry for a lifecycle state.

3.166.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dmc_tcf_activity
- Object type tag: 09

A tcf activity object represents a sequence of actions to execute when an object enters a particular state. TCF activity objects are created when a user identifies one or more actions for a state when defining a lifecycle state in Lifecycle Editor. You cannot create these objects manually. The actions and their parameters are recorded in an XML file that is stored as the content of the tcf activity object.

3.166.2 Properties

The table describes the object properties.

Table 3-162: TCF activity type properties

Property	Datatype	Single or repeating	Description
act_identifier	string(256)	S	Character string identifier that identifies this tcf activity object.
act_template_identifier	string(256)	S	Used internally.

3.167 Transition condition

Records a route case condition expression for an automatic transition of a workflow activity.

3.167.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmc_transition_condition
- Object type tag: 00

A transition condition object records a conditional expression in a route case condition. You cannot create these objects directly. They are created when an addConditionRouteCase method (defined for the IDfActivity interface) is executed by a Foundation Java API at version level 5.3 FCS. This method is executed to save an activity's route case conditions when any one or more of the route cases contains an XPath expression.



Note: When the addConditionRouteCase method is called by a Foundation Java API 5.3 SP1 or later, the method creates routecase condition objects, instead of transition condition objects, to record the expression.

3.167.2 Properties

The table describes the object properties.

Table 3-163: Transition condition type properties

Property	Datatype	Single or repeating	Description
r_aspect_name	string(64)	R	Used internally.
r_attribute_name	string(32)	S	Name of the property referenced in the expression, if any.
r_boolean_value	Boolean	S	Value to be used in the comparison if the datatype identified in r_value_type is Boolean. This is not set if unless the datatype is Boolean.
r_double_value	Double	S	Value to be used in the comparison if the datatype identified in r_value_type is double. This is not set if unless the datatype is double.

Property	Datatype	Single or repeating	Description
r_id_value	ID	S	<p>Value to be used in the comparison if the datatype identified in r_value_type is ID.</p> <p>This is not set if unless the datatype is ID.</p>
r_int_value	integer	S	<p>Value to be used in the comparison if the datatype identified in r_value_type is integer.</p> <p>This is not set if unless the datatype is integer.</p>
r_object_alias	string(128)	S	Name of the package, or manifest values referring to the workflow or work item.
r_relational_op	integer	S	<p>The relation operator in the condition. Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning = • 1, meaning \neq • 2, meaning < • 3, meaning > • 4, meaning \leq • 5, meaning \geq

Property	Datatype	Single or repeating	Description
r_repeating_attr_flag	integer	S	<p>Indicates whether the property named in r_attribute_name is a repeating property and if it is a repeating property, which values to examine when evaluating the condition. Valid values are:</p> <ul style="list-style-type: none"> • -1, meaning the property is not a repeating property • 0, meaning ANY • 1, meaning ALL • 2, meaning FIRST • 3, meaning LAST
r_string_value	string(1024)	S	<p>Value to be used in the comparison if the datatype identified in r_value_type is string.</p> <p>This is not set if unless the datatype is string.</p>
r_time_value	Date	S	<p>Value to be used in the comparison if the datatype identified in r_value_type is Date.</p> <p>This is not set if unless the datatype is Date.</p>
r_value_type	integer	S	Data type of the value in the relational expression. The data type is expressed as an IDfValue constant.
r_xpath_datatype	string(64)	S	<p>The xschema's built-in datatype name.</p> <p>This is set only if the condition includes an XPath expression.</p>

Property	Datatype	Single or repeating	Description
r_xpath_expression	string(1024)	S	An XPath expression. This is set only if the condition includes an XPath expression.
r_xpath_value	string(1024)	S	Literal value used in the XPath transition condition evaluation. The value is in the format in which it is found in the XML document. This is set only if the condition includes an XPath expression.

3.168 Type

Stores structural information about an object type in the repository.

3.168.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dm_type
- Object type tag: 03

A type object stores structural information about an object type in the repository. The object types in Documentum CM Server are themselves represented as types (that is, each type is an object of type dm_type). Documentum CM Server does not allow you to create new objects of type dm_type directly. Instead, when you create a new type or subtype, Documentum CM Server automatically creates an object of type dm_type to describe your new type. To create a new type, use the DQL CREATE TYPE statement.

The values in the repeating properties across one index position represent one property defined for the object type.

3.168.2 Properties

The table describes the object properties.

Table 3-164: Type object type properties

Property	Datatype	Single or repeating	Description
attr_count	integer	S	Number of properties in the type (includes defined and inherited).
attr_identifier	integer	R	Uniquely identifies the property within the type hierarchy. Values start from 1.
attr_length	integer	R	Indicates the length of those properties that are string-valued.
attr_name	string(40)	R	Names of the type's properties.
attr_repeating	Boolean	R	Indicates if the properties are repeating.
attr_restriction	integer	R	Specifies any restrictions on the property. Valid values are: <ul style="list-style-type: none"> • 0, meaning no special restrictions • 1, meaning the property is a non-qualifiable property
attr_type	integer	R	Contains integer values representing the datatype of the properties. Valid values are: <ul style="list-style-type: none"> • 0, Boolean • 1, Integer • 2, String • 3, ID • 4, Date and Time • 5, Double

Property	Datatype	Single or repeating	Description
i_type	integer	S	Used internally.
info	ID	S	Object ID of the type's associated type info object.
name	string(27)	S	Name of the type.
next_attr_identifier	integer	S	Next available property identifier in the type hierarchy.
owner	string(255)	S	Name of the type's owner (creator).
r_object_id	ID	S	Object ID of the type.
r_index_attr	ID	S	Object ID of the index object describing any indexes built on the type's repeating properties.
r_creation_date	DATETIME	S	Date and time when the type (an object of dm_type) was created. Automatically set by the server. This property is Null for types created before an upgrade to OpenText Documentum CM 7.0.

Property	Datatype	Single or repeating	Description
r_modify_date	DATETIME	S	<p>Date and time when the type (an object of dm_type) was last modified. Automatically set by the server.</p> <p>This property is Null for types created before an upgrade to OpenText Documentum CM 7.0 until the types are modified in OpenText Documentum CM 7.0 or later versions.</p> <p>For types created in OpenText Documentum CM 7.0 or later versions, the value of this property equals to the value of the r_creation_date property until the types are modified.</p> <p>When you modify a supertype, the r_modify_date property of the subtype gets updated automatically.</p> <p> Note: Only the following modifications on a type trigger an update of the r_modify_date property:</p> <ul style="list-style-type: none"> • adding a new attribute to the type • dropping an existing attribute from the type

Property	Datatype	Single or repeating	Description
			<ul style="list-style-type: none"> widening an attribute in the type <p>If you change the data dictionary information, such as label_text, or a constraint definition, the r_modify_date property does not change.</p>
shared_parent_name	string(40)	S	<p>For lightweight object types, this is the name of the parent object type. The parent is the type with whose instances the lightweight type instances are sharing properties.</p> <p>For all other types, this is the same as the value in super_name.</p>
start_pos	integer	S	Position of the first non-inherited property.
super_name	string(40)	S	Name of the supertype of the type.
s_index_attr	ID	S	Object ID of the index object describing any indexes built on the type's single-valued properties.
type_category	integer	S	<p>Specifies the category to which the type belongs. Valid values are:</p> <ul style="list-style-type: none"> 0, standard object type 1, aspect property object type 2, shareable object type 4, lightweight object type 8, data table type

Property	Datatype	Single or repeating	Description
views_valid	Boolean	S	Indicates if the views for the type are valid. This is used internally during recovery operations.

3.169 Type info

Stores non-structural information about an object type.

3.169.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_type_info
- Object type tag: 2e

A type info object stores non-structural information about an object type. Storing non-structural information separately from the type's structure definition (the dm_type object) enhances the performance when a type is altered.

3.169.2 Properties

The table describes the object properties.

Table 3-165: Type info type properties

Property	Datatype	Single or repeating	Description
acl_domain	string(255)	S	The domain (owner) of the ACL associated with the type definition. The value is either the name of the user who created the ACL or, for system ACLs, the name of the repository owner.
acl_name	string(32)	S	The object name of the associated ACL.

Property	Datatype	Single or repeating	Description
default_aspects	string(64)	R	List of aspects defined for the object type. These are identified by the name of the aspect as defined in the aspect's object_name property of its dmc_aspect_type object.
default_group	string(255)	S	Default group defined for the type.
default_group_permit	integer	S	Default object-level permission defined at the group level for the type.
default_owner_permit	integer	S	Default object-level permission defined at the owner level for the type.
default_storage	ID	S	Object ID of a storage object of type dm_store. This identifies the default storage type for the contents associated with any object of this type.
default_world_permit	integer	S	Default object-level permission defined at the world level for the type.

Property	Datatype	Single or repeating	Description
ftindex_attrs	string(27)	R	<p>For lightweight object types, contains a list of properties that are indexed. The list does not contain the basic SysObject properties that are indexed by default (acl_name, acl_domain, owner_name, and r_object_type) if the object type has fulltext support.</p> <p>This property is not currently used for any object type except lightweight object types.</p>
i_type_features	integer	S	The value in this property is a bitmask used internally by Documentum CM Server to identify several characteristics of the object type.
locally_managed	Boolean	R	<p>Indicates whether a given property is locally or globally managed if the repository is participating in a federation.</p> <p>The value at each index level corresponds to the property named in attr_names in the type's dm_type object.</p>
r_object_id	ID	S	Object ID of the type info object.

Property	Datatype	Single or repeating	Description
r_orig_declaration	Boolean	R	Indicates whether a given property is defined for the type or inherited. TRUE means it is defined for the type. The value at each index level corresponds to the property named in attr_names in the type's dm_type object.
r_supertype	string(27)	R	List of all supertypes of the type.
r_type_id	ID	S	Object ID of the type object that contains the definition of the type.
r_type_name	string(27)	S	Name of the type represented by the type object identified in r_type_id.
type_cache_vstamp	integer	S	Records the value of the dm9_change_record.type_change_count when the type was changed. When a type is changed, the value of type_change_count is first incremented, and then the new value is written to the type_cache_vstamp property.
type_override	ID	S	Object ID of the dm_aggr_domain object, if any, for the object type.
type_version	integer	S	Version number of the object type. This is incremented when type properties are added, changed, or deleted.

3.170 User

Contains information about a user in the repository.

3.170.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dm_user
- Object type tag: 11

A user object contains information about a user in the repository. The information includes the default permissions of the user, electronic mail address, default folder, and inbox object identifier. You must have Sysadmin or Superuser user privileges to create or drop a user or to activate or deactivate a user.

3.170.2 Properties

The table describes the object properties.

Table 3-166: User type properties

Property	Datatype	Single or repeating	Description
acl_domain	string(255)	S	Specifies the domain (owner) of the ACL associated with the user. The value is either the user who created the ACL or, for system ACLs, the name of the repository owner.
acl_name	string(32)	S	The object name of the associated ACL.
alias_set_id	ID	S	Object ID of the alias set object representing the user-level default alias set.

Property	Datatype	Single or repeating	Description
client_capability	integer	S	<p>Indicates what level of use is expected of the user. Valid values are:</p> <ul style="list-style-type: none"> • 0 or 1, for consumer • 2, for contributor • 4, for coordinator • 8, for system administrator <p>The default value is 0.</p>
deactivated_ip_addr	string(64)	S	Reserved for future use.
deactivated_utc_time	Date	S	Reserved for future use.
default_folder	With Oracle: string(740) With SQL Server: see Description With PostgreSQL: string(450)	S	<p>Specifies the default folder of the user.</p> <p>The default is Temp.</p> <p>The length of the default folder property varies for SQL Server customers:</p> <ul style="list-style-type: none"> • 450: For a new SQL Server repository. • 765: For an upgraded SQL Server repository not using Unicode.
description	string(255)	S	User-defined description of the user.

Property	Datatype	Single or repeating	Description
failed_auth_attempt	integer	S	<p>Number of unsuccessful authentication attempts on behalf of the user.</p> <p>Except for the installation owner, this value is set to 0 when a user is created. The default value for the installation owner is -1.</p> <p>Setting this value to -1 for any user disables the feature for that user.</p> <p>If the feature is enabled for a user, the value is reset to 0 whenever the user is authenticated successfully.</p>
first_failed_auth_utc_time	Date	S	Reserved for future use.
globally_managed	Boolean	S	<p>Indicates whether the user object is managed globally or locally. The default is FALSE, meaning that it is locally managed.</p> <p>Requires at least Sysadmin privileges to change.</p>
group_def_permit	integer	S	Default group permit. This property is used to assign object-level permission at the group level to any object the user creates if no permission at that level is explicitly assigned.
home_docbase	string(120)	S	The home repository of the user.

Property	Datatype	Single or repeating	Description
last_login_utc_time	Date	S	<p>If <code>dfc.session.pooling=f</code> also in the <code>dfc.properties</code> file, <code>last_login_utc_time</code> is the time, in UTC format, of the last login of the user to a repository.</p> <p>If <code>dfc.session.pooling=true</code>, you cannot use this property to determine the last login time of the user. As an alternative, it is possible to use the <code>dm_connect</code> audit event to find the last login of a user.</p> <p>This property is not set if the user is the <code>dmadmin</code> user or if the value in the property has been set to <code>NULLDATE</code>.</p> <p> Note: Only a superuser can set this value to <code>NULLDATE</code>.</p> <p>The current and default behavior is:</p> <ul style="list-style-type: none"> When you create the user, the <code>last_login_utc_time</code> attribute values is set to <i>current time</i>. When the user logs in, the <code>last_login_utc_time</code> value gets updated with the actual user's login time only if the existing value is not <code>NULLDATE</code>.

Property	Datatype	Single or repeating	Description
			<p>A flag, <code>dm_last_login_utc_time_behavior</code> is introduced in <code>dm_docbase_config</code>'s <code>r_module_name</code>, to support the 53 behavior.</p> <p>If there is a pair <code>(dm_last_login_utc_time_behavior, 53)</code> in <code>(r_module_name, r_module_mode)</code>, then the behavior is as follows:</p> <ul style="list-style-type: none"> • When you create the user, <code>last_login_utc_time</code> attribute values is set to <code>NULLDATE</code>. • When the user logs in, irrespective of the existing value (either <code>NULLDATE</code> or any other date value), <code>last_login_utc_time</code> gets updated with the actual user's login time.
<code>owner_def_permit</code>	integer	S	Default owner permit. This attribute is used to assign object-level permission at the owner level to any object the user creates if no permission at that level is explicitly assigned.
<code>r_has_events</code>	Boolean	S	Specifies whether someone has registered the user for auditing.

Property	Datatype	Single or repeating	Description
r_is_group	Boolean	S	Indicates if the user represents a group or an individual user.
r_modify_date	date	S	Time and date of the last change to the user object.
r_object_id	ID	S	Object ID of the user.
restricted_folder_ids	ID	R	<p>Object IDs of the cabinets or folders which the user can access. The user can access these objects and their subfolders. If set, the user can access only these folders.</p> <p>This property is a local property, not a global property.</p>
root_log_dir	string(255)	S	<p>Name of the parent user log directory when the number of users in the repository exceeds 31,998 on Linux platforms.</p> <p>This property only works in Linux Ext3 file systems.</p> <p>This property is empty when the number of users is less than 31,998. In this case, the user log directories are located in the following default path:</p> <p>\$DOCUMENTUM/dba/log /<docbase_id></p>
user_address	string(80)	S	The electronic mail (email) address of the user. This property is required.

Property	Datatype	Single or repeating	Description
user_admin	string(255)	S	Specifies who is allowed to modify the user object. Valid values are an individual user name or a group name.
user_db_name	string(32)	S	User name of the user in the underlying RDBMS. This value must consist of ASCII characters.
user_delegation	string(255)	S	The name of a user to whom to delegate work items.
user_global_unique_id	string(255)	S	Reserved for future use.
user_group_name	string(255)	S	The default group. This value is used when the user creates an object as part of the determination of the default group of the object.
user_initials	string(16)	S	Reserved for future use.
user_ldap_dn	string(512)	S	The Distinguished Name of the user in LDAP. This property is deprecated.
user_login_domain	string(255)	S	Name of the Windows domain against which this user is authenticated. This value can be blank.

Property	Datatype	Single or repeating	Description
user_login_name	string(255)	S	<p>Name used to authenticate the user. This property must be set.</p> <p>If a domain is not required in the repository, user_login_name must be unique within the repository.</p> <p>If a domain is required, the combination of user_login_name and user_login_domain must be unique.</p>
user_name	string(255)	S	<p>Documentum CM Server user name. This value can be an individual user or a group name. It must be unique among the user and group names in the repository.</p> <p>The name must consist of characters compatible with the server_os_codepage of the Documentum CM Server.</p> <p>This property is required.</p>

Property	Datatype	Single or repeating	Description
user_os_domain	string(15)	S	<p>Windows domain name associated with user. Valid values are:</p> <ul style="list-style-type: none"> • * (asterisk) or blank (no value): Documentum CM Server authenticates the user using operating system name and the domain specified in the connection request. If no domain is included in the connection request, the server uses the default domain defined in the user_auth_target key of the server.ini file. • <domain name>: Documentum CM Server authenticates the user against the specified domain.
user_os_name	string(32)	S	<p>The name of the operating system account of the user, if any.</p> <p>This value must consist of ASCII characters if specified.</p>
user_password	string(256)	S	<p>Encrypted password for the user. This value is set only if the user_source property of the user is set to <i>inline password</i>.</p> <p>When displayed, it displays as 16 asterisks (*).</p>

Property	Datatype	Single or repeating	Description
user_privileges	integer	S	<p>Specifies the user privileges of the user. This value is the sum of the values corresponding to the user privileges defined for the user. For example, if a user has Create Type and Create Cabinet user privileges, the value is 3.</p> <p>The privileges and their values are:</p> <ul style="list-style-type: none">• None (0)• Create Type (1)• Create Cabinet (2)• Create Group (4)• Sysadmin (8)• Superuser (16) <p>For more information about the user_privileges property, see <i>OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS250400-AGD)</i>.</p>

Property	Datatype	Single or repeating	Description
user_source	string(16)	S	<p>Indicates the authentication source for this user. Valid values are:</p> <ul style="list-style-type: none"> • <i>OTDS</i>, meaning that the user is authenticated against OTDS. • <i>unix only</i>, meaning the user is authenticated by standard Linux mechanism; the domain isn't used. • <i>domain only</i>, meaning the user is authenticated against the Windows domain; the Linux password file is not used. • <i>unix first</i>, meaning the user is authenticated first by the standard Linux mechanism. If that fails, the user is authenticated against the Windows domain. • <i>domain first</i>, meaning that the user is authenticated first against the Windows domain. If that fails, the user is authenticated using the standard Linux mechanism. • <i>inline password</i>, meaning that the user is authenticated against the password stored

Property	Datatype	Single or repeating	Description
			<p>in the user_password property.</p> <ul style="list-style-type: none">• <i>plugin_identifier</i>, meaning that the user is authenticated using the plug-in identified by the identifier. <p> Note: The Linux- and domain-related values are effective only if an auth config object exists in the repository.</p>

Property	Datatype	Single or repeating	Description
user_state	integer	S	<p>Indicates the activation state of the user. Valid values are:</p> <ul style="list-style-type: none"> • 0, indicating a user is currently active repository user. Active users are able to connect to the repository. • 1, indicating a user is not currently active in the repository. Inactive users are unable to connect to the repository. The user automatically moves to ACTIVE state after auth_deactivation_interval minutes, if auth_deactivation_interval is greater than zero. The user is automatically assigned to INACTIVE state by Documentum CM Server if the user exceeds max_auth_attemp failure attempts within auth_failure_interval minutes. • 2, indicating a user unable to connect to the repository. • 3, indicating a user is inactive and unable to connect to the repository. The user moves to locked and

Property	Datatype	Single or repeating	Description
			inactive states, if a locked user exceeds max_auth_attemp failure attempts within auth_failure_inter val minutes. Value of user_state overrides the values of workflow_disabled meaning that if user_state is active then only the workflow_disabled value is considered.
user_web_page	string(255)	S	Reserved for future use.

Property	Datatype	Single or repeating	Description
user_xprivileges	integer	S	<p>The extended user privileges of the user. The value is the sum of the values corresponding to the extended user privileges defined for the user. For example, if a user has Config Audit and Purge Audit privileges, the value is 24.</p> <p>The privileges and their values are:</p> <ul style="list-style-type: none"> • 1, Config dm_critical_event _receiver_role. This means adding any new user to this role. • 2, Purge dm_critical_event _receiver_role. Reserved for future use. • 4, Purge dm_critical_event _receiver_role. Reserved for future use. • 8, Config Audit • 16, Purge Audit • 32, View Audit
workflow_disabled	Boolean	S	<p>Indicates whether the user is available for work item assignment. Valid values are:</p> <ul style="list-style-type: none"> • 0, available • 1, not available <p>The default is 0 (available).</p>

Property	Datatype	Single or repeating	Description
world_def_permit	integer	S	Default world permit. This value is used to assign object-level permission at the world level to any object the user creates if no permission at that level is explicitly assigned.

3.171 Validation descriptor

Stores configuration information used by Foundation Java API and Accelerated Content Services or Branch Office Caching Services servers.

3.171.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_validation_descriptor
- Object type tag: 08

Validation descriptor objects record configuration information that is shared between Foundation Java API and Accelerated Content Services or Branch Office Caching Services servers. These objects are created and managed internally by Documentum CM Server.



Caution

Do not modify this type or instances of this type.

3.171.2 Properties

The table describes the object properties.

Table 3-167: Validation descriptor type properties

Property	Datatype	Single or repeating	Description
descriptor_version	string(16)	S	Used internally.
parameter_names	string(255)	R	Used internally.

3.172 Validation module

Represents a business module that corresponds to a Docbasic expression defined in a func expr object.

3.172.1 Description

- Supertype: Module
- Subtypes: None
- Internal name: dmc_validation_module
- Object type tag: 0b

A validation module represents the Java equivalent of the Docbasic expressions identified in func expr objects. Each validation module stores the Java equivalents for all Docbasic expressions defined for check constraints or for conditional value assistance for a particular object type. A validation module object is associated with the equivalent func expr objects by a relationship whose name is dmc_expr_to_module. The validation module is related to the dmc_jar object that contains the compiled code through a relationship named dmc_module_to_jar.

These objects are created only when a Docbasic expression is manually converted to a Java expression. OpenText Documentum CM 6 and later does not require manual conversion, as the conversion happens at runtime automatically. Manual conversion is required only for clients prior to OpenText Documentum CM 6.0.

3.172.2 Properties

The table describes the object properties.

Table 3-168: Validation module type properties

Property	Datatype	Single or repeating	Description
expr_code_object_id	ID	S	<p>Object ID of the dmi_expr_code object that contains the original Docbasic expressions file and P-code file for the expressions.</p> <p>The expressions are compiled into one file and the P-code file generated from that file. Both the source file and the P-code file are stored as content of the expr code object.</p>
implementation_enabled	Boolean	S	<p>Specifies whether Java evaluation is enabled for the expressions implemented by this module.</p> <p>T means that Java evaluation is enabled; F means that Java evaluation is not enabled.</p> <p>The value in this property takes precedence over the expr_enabled setting in the associated dmc_validation_relation object.</p>
referenced_type_name	string(27)	R	Name of the object type to which the expression is bound. Currently, only one value in this property is supported.

3.173 Validation relation

Relates a func expr object to a validation module object.

3.173.1 Description

- Supertype: Relation
- Subtypes: None
- Internal name: dmc_validation_relation
- Object type tag: 37

A validation relation object relates a func expr object to the validation module that represents the Java equivalent of the Docbasic expression in the func expr object. The relation_name value for all validation relation objects is dmc_expr_to_module.

3.173.2 Properties

The table describes the object properties.

Table 3-169: Validation relation type properties

Property	Datatype	Single or repeating	Description
expr_enabled	Boolean	S	<p>Specifies whether Java evaluation is enabled for the expression represented by the related validation module. T means Java evaluation is enabled; F means that Java evaluation is not enabled.</p> <p>The setting in this property can be overridden by the setting of the implementation_enabled property in the dmc_validation_module object.</p>

Property	Datatype	Single or repeating	Description
has_implementation	Boolean	S	<p>Indicates whether actual Java code exists for the expression.</p> <p>This is set to F if the Docbasic expression was not successfully migrated to Java code for the validation module.</p> <p>If this is F, then expr_enabled is F and cannot be reset to T.</p>

3.174 Value assist

Describes the value assistance provided through the data dictionary for an property.

3.174.1 Description

- Supertype: Persistent Object
- Subtypes: Value List, Value Query, Value Func
- Internal name: dm_value_assist
- Object type tag: 5a

A value assist object describes the value assistance provided through the data dictionary for an property. Value assist objects are created and managed by Documentum CM Server and cannot be created by users.

3.174.2 Properties

The table describes the object properties.

Table 3-170: Value assist type properties

Property	Datatype	Single or repeating	Description
complete_list	Boolean	S	Indicates whether the values returned by value assistance are the only acceptable values for the property. TRUE means that they are the only acceptable values. FALSE means that other values are acceptable.
domain_type	integer	S	Indicates the datatype for the values returned by value assistance. Valid values are: <ul style="list-style-type: none"> • 0, meaning Boolean • 1, meaning Integer • 2, meaning String • 3, meaning ID • 4, meaning Time/date • 5, meaning Double
object_name	string(32)	S	Currently unused.
parent_id	ID	S	Object ID of the aggr domain object to which the property belongs.
use_as_constraint	Boolean	S	Currently unused.
value_estimate	integer	S	The number of values expected to be returned by value assistance.

3.175 Value func

Describes the procedural form of value assistance.

3.175.1 Description

- Supertype: Value Assist
- Subtypes: None
- Internal name: dm_value_func
- Object type tag: 5d

A value func object describes the procedural form of value assistance. Value func objects are created and managed by Documentum CM Server and cannot be created by users.

3.175.2 Properties

The table describes the object properties.

Table 3-171: Value func type properties

Property	Datatype	Single or repeating	Description
func_expression	ID	S	Object ID of the func expr object representing the user function to call.
value_separator	string(1)	S	Defines the character used as a separator in the value list returned by the user function. The default is a comma (,).

3.176 Value list

Contains the valid values for the list form of value assistance.

3.176.1 Description

- Supertype: Value Assist
- Subtypes: None
- Internal name: dm_value_list
- Object type tag: 5b

A value list object contains the valid values for the list form of value assistance. Value list objects are created and managed by Documentum CM Server and cannot be created by users.

3.176.2 Properties

The table describes the object properties.

Table 3-172: Value list type properties

Property	Datatype	Single or repeating	Description
valid_values	string(255)	R	<p>List of string values representing literals of the appropriate data type. Each literal must be unique within the list.</p> <p>The list values are displayed to users in the order in which they appear in the property.</p>

3.177 Value query

Contains information for the query form of value assistance.

3.177.1 Description

- Supertype: Value Assist
- Subtypes: None
- Internal name: dm_value_query
- Object type tag: 5c

A value query object contains information for the query form of value assistance. Value query objects are created and managed by Documentum CM Server and cannot be created by users.

3.177.2 Properties

The table describes the object properties.

Table 3-173: Value query type properties

Property	Datatype	Single or repeating	Description
allow_caching	Boolean	S	Indicates whether the queries defined for value assistance can be cached. TRUE means that the queries can be cached. FALSE means that the query should be rerun each time it is needed.
query_attribute	string(32)	S	The property from the query's selected values list that will provide the data to be displayed. The default is the first selected value.
query_string	string(255)	R	The DQL SELECT statement that returns the value assistance values. If it is longer than 255 characters, the first 255 characters appear in query_string[0], the second 255 characters in query_string[1], and so on. The continuation character is an underscore (_).

3.178 ViPR store

Represents a ViPR content storage area.

3.178.1 Description

- Supertype: Store
- Subtypes: none
- Internal name: dm_vipr_store
- Object type tag: inherited from Store

ViPR store object represents a ViPR content storage area. The properties in ViPR store objects may be changed only by users with Sysadmin or Superuser user privileges.

3.178.2 Properties

ViPR store objects inherit all of their properties from their supertype, the Store type.

3.179 Vstamp

Used internally by the system, at start-up, to validate the consistency of the server and repository.

3.179.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_vstamp
- Object type tag: 1e

A vstamp object is used internally by the system, at startup, to validate the consistency of the server and repository.

3.179.2 Properties

The table describes the object properties.

Table 3-174: Vstamp type properties

Property	Datatype	Single or repeating	Description
i_application	string(64)	S	Name of the client program.

Property	Datatype	Single or repeating	Description
i_stamp	integer	S	Value used internally to validate repository consistency.

3.180 WF attachment

Describes an attachment to a workflow.

3.180.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_wf_attachment
- Object type tag: 00

A wf attachment object describes an object added at runtime to a workflow or workitem as an attachment.

3.180.2 Properties

The table describes the object properties.

Table 3-175: WF attachment type properties

Property	Datatype	Single or repeating	Description
r_component_id	ID	S	Object Id of the object attached to the workflow.
r_component_name	string(80)	S	Name of the object attached to the workflow.
r_component_type	string(40)	S	Object type (actual or supertype) of the object attached to the workflow.
r_creation_date	Date	S	Date and time the attachment was attached to the workflow.
r_creator_name	string(255)	S	Name of the user who attached the object to the workflow.

Property	Datatype	Single or repeating	Description
r_workflow_id	ID	S	Object ID of the workflow to which the object is attached.

3.181 WF package report parent

The parent type of workflow package report types.

3.181.1 Description

- Supertype: None
- Subtypes: None
- Internal name: dmc_wpr_parent
- Object type tag: 00

3.181.2 Properties

The table describes the object properties.

Table 3-176: WF package report parent type properties

Property	Datatype	Single or repeating	Description
act_name	string(128)	S	Contains the name of the activity where the package is defined.
act_sequenceno	integer	S	Contains the sequence number of the activity instance.
component_name	string(255)	S	Contains the name of the component contained in the package.
package_name	string(128)	S	Contains the name of the package.
performer_name	string(32)	S	Contains the name of the performer who completes the activity instance.
process_id	ID	S	Contains the ID of the process (dm_process object) where the package is defined.

Property	Datatype	Single or repeating	Description
process_name	string(255)	S	Contains the name of the process defined in process_id.
time_stamp_utc	Time	S	Contains the time the report entry is created.
type_info_id	ID	S	Contains the ID of the type info object (dmc_wf_package_type_info object) of the type used in the workflow package.
workflow_id	ID	S	Contains the ID of the workflow instance.
workitem_id	ID	S	Contains the ID of the work item instance.

3.182 WF package schema

Stores the URI of a workflow package schema.

3.182.1 Description

- Supertype: Relation
- Subtypes: None
- Internal name: dmc_wf_package_schema
- Object type tag: 37

A wf package schema object stores the URI of a schema associated with a package whose component is an XML file. It is used to validate any XPath expression referencing that file in a transition condition for an activity. WF package schema objects are created when the package is defined. The object type is installed by a script when Documentum CM Server is installed.

3.182.2 Properties

The table describes the object properties.

Table 3-177: WF package schema type properties

Property	Datatype	Single or repeating	Description
schema_uri	string(255)	S	URI of a schema.

3.183 WF package skill

Associates a skill level required for a workflow package with a workflow.

3.183.1 Description

- Supertype: Relation
- Subtypes: None
- Internal name: dmc_wf_package_skill
- Object type tag: 37

A wf package skill object identifies the skill level a user must have to acquire or be assigned the task associated with a particular package. This feature is effective only when applied to tasks on workqueues. WF package skill objects are created at runtime and associated with the workflow.



Caution

Do not modify this object type nor instances of the object. These objects are created, maintained, and destroyed internally, as needed.

3.183.2 Properties

The table describes the object properties.

Table 3-178: WF package skill type properties

Property	Datatype	Single or repeating	Description
child_id	ID	S	This is not set.
child_label	string(32)	S	There is no label set for instances of this type.
Description	string(250)	S	There is no description provided for instances of this type.

Property	Datatype	Single or repeating	Description
package_name	string(128)	S	Name of the package that requires the skill level identified in skill_level.
parent_id	ID	S	Object ID of the workflow with which the package is associated
permanent_link	Boolean	S	Set to T (TRUE).
relation_name	string(32)	S	Set to dmc_wf_package_skill.
skill_level	integer	S	An integer value identifying the required skill level.

3.184 WF package type info

Specifies a type info object for a type used in a workflow package

3.184.1 Description

- Supertype: dm_sysobject
- Subtypes: None
- Internal name: dmc_wf_package_type_info
- Object type tag: 08

3.184.2 Properties

The table describes the object properties.

Table 3-179: WF package type info type properties

Property	Datatype	Single or repeating	Description
attr_is_filterable	integer	R	Indicates whether the attribute defined at the corresponding index of reportable_attribute can be filtered or not.
package_type_name	string(27)	S	Contains the name of the data type used in workflow package.

Property	Datatype	Single or repeating	Description
report_type_name	string(27)	S	Contains the name of the report type corresponds to the type defined in package_type_name.
reportable_attribute	string(40)	R	Contains names of reportable attributes in the type defined in package_type_name.

3.185 WF timer

Records the configuration of a timer for a workflow activity.

3.185.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_wf_timer
- Object type tag: 00

A wf timer object describes a timer for a workflow activity. The wf timer objects are created automatically when the timer is instantiated. The object type is installed by a script when Documentum CM Server is installed.

3.185.2 Properties

The table describes the object properties.

Table 3-180: WF timer type properties

Property	Datatype	Single or repeating	Description
i_calendar_status	integer	S	Indicates whether this timer has a calendar associated with it. If so, the due date needs to be recalculated. Valid values are: <ul style="list-style-type: none"> • 1, meaning yes • 0, meaning no
r_act_id	ID	S	Object ID of the activity for which this timer is defined.

Property	Datatype	Single or repeating	Description
r_act_name	string(128)	S	Name of the activity for which this timer is defined.
r_act_seqno	integer	S	Sequence number of the activity instance.
r_action_id	ID	S	Object ID of the module config object associated with the timer.
r_action_index	integer	S	The index to the r_pre_timer_action or r_post_timer_action property of the activity.
r_calendar_id	ID	S	Reserved for future use.
r_is_initialized	integer	S	Indicates whether the timer is initialized or not. Valid values are: <ul style="list-style-type: none"> • 0, meaning not initialized • 1, meaning initialized
r_timer	Date	S	Absolute date and time at which to trigger the timer.
r_timer_name	string(64)	S	Name of the timer object as defined in the dm_activity object.
r_timer_type	integer	S	Specifies what kind of timer this is. Valid values are: <ul style="list-style-type: none"> • 0, meaning a pre-timer • 1, meaning a post-timer • 2, meaning a suspend-timer
r_workflow_id	ID	S	Object ID of the workflow that contains the activity instance.

3.186 Work item

Stores information about a task for a human or automatic performer.

3.186.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmi_workitem
- Object type tag: 4a

Work items are generated by Documentum CM Server from an activity object. Users cannot create work items. Users can modify only the following properties: a_held_by, a_wq_doc_profile, a_wq_flag, a_wq_name, and a_wq_policy_id.

3.186.2 Properties

The table describes the object properties.

Table 3-181: Work item type properties

Property	Datatype	Single or repeating	Description
a_control_instruction	string(32)	S	Control instruction.
a_held_by	string(255)	S	Name of the user (up to 255 characters) who has acquired the task.
a_wq_doc_profile	string(64)	S	Name of the work queue doc profile associated with this task.
a_wq_flag	integer	S	Indicates whether the task is pushed to a queue member or pulled by a queue member. Valid values are: <ul style="list-style-type: none"> • 0, meaning the task is pulled by a queue member • 1, meaning the task is pushed to a queue member The default is 1.

Property	Datatype	Single or repeating	Description
a_wq_name	string(32)	S	Name of the work queue to which this work item is assigned if the task is assigned to a work queue. For tasks generated by automatic activities, this may be set to an error message if Documentum CM Server cannot process the task. For more information about the a_wq_name property, see <i>OpenText Documentum Content Management - Server Fundamentals Guide (EDCCS250400-GGD)</i> .
a_wq_policy_id	ID	S	Object ID of the work queue policy that controls how the work item is handled on the queue.
r_act_def_id	ID	S	Refers to the activity definition object.
r_act_seqno	integer	S	Records the sequence number of the activity in which the package is being handled.
r_auto_method_id	ID	S	Contains the object ID of an application to be invoked (a dm_method instance).
r_creation_date	datetime	S	Records date and time of when a work item is generated.
r_due_date	datetime	S	Indicates the date and time when a work item is expected to complete.

Property	Datatype	Single or repeating	Description
r_exec_launch	Boolean	S	Indicates whether the work item is currently executing. TRUE means the work item is executing; FALSE means it is not executing.
r_exec_os_error	string	S	Contains the operating system error string, if any.
r_exec_result_id	ID	S	Contains the document ID of the saved results of the application execution.
r_exec_retried_count	integer	S	For automatic tasks, the number of times execution has been attempted.
r_exec_time_out	Boolean	S	Indicates if an execution times out.
r_ext_performer	string(255)	R	Lists new performers to repeat the same activity. If a group name, only one work item is generated and any group member can acquire the work item.
r_handling_instruction	integer	S	Used internally by the workflow agent to handle automatic tasks. Valid values are: <ul style="list-style-type: none"> • 0, meaning handle in the regular manner • 1, meaning TBD

Property	Datatype	Single or repeating	Description
r_launch_timeout	Date	S	Records the date and time at which a work item execution times out. This is set when the work item begins execution if the work item is generated by an automatic activity. Otherwise, its value is NULLDATE.
r_next_retry_date	Date	S	Time at which a failed task will be retried.
r_output_port	string(16)	R	Allows a performer to specify a set of output ports by their unique names.
r_performer_name	string(255)	S	Contains the name of an activity performer, or the owner name of the method if this is an automatic work item.
r_priority	integer	S	Represents the priority assigned by a performer or an application. The default value comes from the activity_priority property of dm_process of the activity by which the work item is generated.
r_queue_item_id	ID	S	Refers to a peer dmi_queue_item object.

Property	Datatype	Single or repeating	Description
r_runtime_state	integer	S	<p>Contains the current state of the work item. Valid value are:</p> <ul style="list-style-type: none"> • 0, meaning dormant • 1, meaning acquired • 2, meaning finished • 3, meaning paused • 4, meaning Dpaused (a work item in the dormant state is paused) • 5, meaning Apaused (a work item in the acquired state is paused) • 6, meaning Ppaused (a work item in the paused state is paused) • 7, meaning Faulted <p>Values 4-6 occur when a workflow containing the work item is halted.</p>
r_target_task_id	ID	S	For fault-handling tasks, this is the object ID of the automatic activity for which this fault-handling task was created.
r_workflow_id	ID	S	Contains the object ID of the workflow that generated this item.
return_value	integer	S	Contains the returned value set by a performer or an application.

Property	Datatype	Single or repeating	Description
user_cost	double	S	Actual user cost spent to complete the work item. The default value is 0.
user_time	integer	S	Actual amount of time spent by the user to complete the work item. The default is 0.

3.187 Work queue

Represents a work queue for work items generated from a workflow.

3.187.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmc_workqueue
- Object type tag: 00

A work queue object represents a work queue for workflow tasks. The workqueue object type is installed by a script when Documentum CM Server is installed. However, work queues (instances of the type) are created and managed using Webtop.

3.187.2 Properties

The table describes the object properties.

Table 3-182: Work queue type properties

Property	Datatype	Single or repeating	Description
skill_info_ids	ID	R	Object IDs of the wq skill info objects that identify the skills required to complete the tasks in this work queue.

Property	Datatype	Single or repeating	Description
wq_category_id	ID	S	Object ID of the workqueue category object representing the category to which this work queue belongs.
wq_name	string(32)	S	Name of the work queue.
wq_policy_id	ID	S	Object ID of the workqueue policy used to handle work items in this queue.

3.188 Work queue category

Defines a category for a work queue.

3.188.1 Description

- Supertype: Folder
- Subtypes: None
- Internal name: dmc_workqueue_category
- Object type tag: 0b

A work queue category object represents a category of work queues.

The object type is installed by a script when Documentum CM Server is installed. However, instances of the type are created and managed through Webtop when work queues are set up and managed.

3.188.2 Properties

The object type has no properties defined for it. It inherits all its properties from its supertype, dm_folder. It uses only the object_name, acl_name, and acl_domain properties. The object_name property stores the work queue category name. The acl_name and acl_domain properties are set to specify the ACL named *Work Queue User Default ACL* for all work queue category objects.

3.189 Work queue doc profile

Stores information about a particular kind of document.

3.189.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmc_workqueue_doc_profile
- Object type tag: 00

A work queue doc profile contains information that describes a particular kind of document. The objects are used to manage work items placed on work queues. Work queues, and consequently, work queue doc profiles, are created using Webtop. The object type is installed using a script at the time Documentum CM Server is installed.

3.189.2 Properties

The table describes the object properties.

Table 3-183: Work queue doc profile type properties

Property	Datatype	Single or repeating	Description
doc_profile_name	string(64)	S	Name of the doc profile.
owner_name	string(255)	S	Name of the user who owns the work queue doc profile object.
wq_name	string(32)	R	Names of the work queues to which documents that use this profile may be assigned.
wq_policy_id	ID	R	Object IDs of work queue policy objects. The work queue policy object identified at a particular index position is associated with the work queue specified at the corresponding index position in wq_name.

3.190 Work queue policy

Defines configuration information for a work queue.

3.190.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmc_workqueue_policy
- Object type tag: 00

A work queue policy object defines configuration parameters for handling a task in a workqueue. The parameters control how the items are handled. Each work queue has one associated work queue policy. If a document associated with a task has a defined work queue policy, that policy overrides the work queue's policy.

The work queue policy object type is installed by a script when Documentum CM Server is installed.

3.190.2 Properties

The table describes the object properties.

Table 3-184: Work queue policy type properties

Property	Datatype	Single or repeating	Description
calendar_id	ID	S	Reserved for future use.
increment_priority	integer	S	Value by which the task priority is incremented.
increment_priority_method	string(256)	S	Reserved for future use.
increment_priority_mode	integer	S	Reserved for future use.
initial_priority	integer	S	Initial priority value of tasks in the work queue.
max_priority	integer	S	Maximum allowed priority value for an unfinished task on the queue.
max_threshold	ID	S	Maximum number of unfinished tasks allowed in the queue.

Property	Datatype	Single or repeating	Description
owner_name	string(255)	S	Name of the user who owns this work queue policy object.
percent_quality_check	integer	S	Value, interpreted as percentage, that determines whether the quality assurance is performed on the task when it is completed.
policy_name	string(255)	S	Name of the work queue policy.
policy_type	integer	S	Specifies whether the policy defines the policy of a work queue or a document. Valid values are: <ul style="list-style-type: none">• 0, meaning it is a work queue policy• 1, meaning it is a document policy

3.191 Work queue user profile

Records information about a work queue user.

3.191.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmc_workqueue_user_profile
- Object type tag: 00

A work queue user profile describes a user who performs tasks taken from a work queue. The information includes the user's skill level. The user profiles are created when the work queue is created.

The object type is installed by a script when Documentum CM Server is installed.

3.191.2 Properties

The table describes the object properties.

Table 3-185: Work queue user profile type properties

Property	Datatype	Single or repeating	Description
doc_profile_name	string(64)	R	Names of doc profile objects.
owner_name	string(255)	S	Name of the owner of this workqueue user profile object.
skill_level	integer	R	Defines skill levels for the user. The level identified at a particular index position applies to the workqueue and doc profile identified in the corresponding index positions in workqueue_name and doc_profile_name.
user_name	string(255)	S	Name of the user.
workqueue_name	string(32)	R	Names of the work queues.

3.192 Workflow

Contains the runtime information about a workflow.

3.192.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dm_workflow
- Object type tag: 4d

A workflow object contains the runtime information about a workflow.

3.192.2 Properties

The table describes the object properties.

Table 3-186: Workflow type properties

Property	Datatype	Single or repeating	Description
correlation_identifier	string(128)	S	A string value that uniquely identifies the workflow instance.
i_next_act_group_seqno	integer	S	Indicates the next sequence number of an activity group instance. For internal use only.
i_next_seqno	integer	S	Indicates the next sequence number of an activity instance.
i_performer_flag	integer	R	Indicates special run-time conditions such as extension.
initiate_act	string(128)	S	Name of the Initiate activity, if any, from which the workflow was initiated.
instructions	string(255)	R	User-defined string displayed to users who are using the sendToDistributed workflow. Applications can reference this property. It must be set before the first Save operation on the workflow object.
object_name	string(128)	S	Contains the workflow name.
parent_act_name	string(128)	S	Name of the activity in the workflow identified in parent_id from which this workflow was started.

Property	Datatype	Single or repeating	Description
parent_act_seqno	integer	S	Sequence number of the activity identified in parent_act_name.
parent_id	ID	S	Object ID of the workflow that contains the activity that started this workflow.
process_id	ID	S	Contains the object ID of process definition (the dm_process object) on which this workflow is based.
r_act_def_id	ID	R	Contains the object ID of the activity definitions (dm_activity objects) included in the process definition identified in process_id.
r_act_errno	integer	R	Records the failing operation, if any. Used for error recovery.
r_act_name	string(128)	R	Contains the activity identifier as defined in the dm_process object.
r_act_seqno	integer	R	Contains the unique sequence number of an activity instance.
r_act_state	integer	R	Records the current state of the activity. Valid values and their corresponding states are: <ul style="list-style-type: none"> • 0, meaning dormant • 1, meaning active • 2, meaning finished • 3, meaning halted • 4, meaning failed

Property	Datatype	Single or repeating	Description
r_alias_set_id	ID	S	Records the object ID of the alias set used to resolve performer aliases when the workflow is created. This is a runtime copy of the alias set identified in perf_alias_set_id of the dm_process object.
r_complete_witem	integer	R	Records the number of completed work items.
r_creator_name	string(255)	S	Indicates the creator. Automatically set by the server.
r_last_performer	string(255)	R	Contains the activity performer who last marks his or her own work item as complete. Before any work item of an activity instance completes, this may contain the last performer of the previous activities.
r_last_witem_id	ID	R	Records the ID of the work item that was most recently marked as complete.
r_perf_act_name	string(128)	R	Contains the names of activities whose performers were chosen at workflow initiation or upon completion of another activity. An activity name appears once for each performer of the activity. The activity's performer appears at the corresponding index position in r_performers.

Property	Datatype	Single or repeating	Description
r_performers	string(255)	R	<p>Contains the name of a user or group chosen at workflow initiation or upon completion of another activity as a performer for the activity at the corresponding index position in r_perf_act_name.</p> <p>If there are multiple performers for an activity, each performer appears at a different index position, and r_perf_act_name records the name of the activity in the corresponding index positions.</p>
r_post_timer	Date	R	This property is obsolete in version 5.3. If the workflow and activity were started prior to upgrading to 5.3, this value is the absolute date and time when the activity should finish.
r_pre_timer	Date	R	This property is obsolete in version 5.3. If the workflow was started prior to upgrading to 5.3, this value is the absolute date and time when the activity should start.
r_repeate_invoke	Boolean	R	Indicates whether this activity can be triggered multiple times.

Property	Datatype	Single or repeating	Description
r_runtime_state	integer	S	Indicates the current state of the workflow. Values are: <ul style="list-style-type: none"> • 0, meaning dormant • 1, meaning running • 2, meaning finished • 3, meaning halted • 4, meaning terminated
r_start_date	date	S	Represents the start time of the instance, set by the server when the workflow enters the running state.
r_target_act_seqno	integer	R	For fault handling activities, this is the sequence number of the activity of the failed automatic activity for which this fault handling activity was invoked. For all other activity types, this is set to -1.
r_total_witem	integer	R	Records the total number of generated work items.
r_trigger_revert	integer	R	Records the number of revert ports triggered (either 0 or 1).
r_trigger_input	integer	R	Records the number of input ports triggered.
r_trigger_thresh	integer	R	Indicates the triggering threshold.
supervisor_name	string(255)	S	Indicates the workflow supervisor. The default is the creator.

3.193 Wq skill info

Describes a skill required for the performance of a task on a workqueue.

3.193.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmc_wq_skill_info
- Object type tag: 00

An object of type wq skill info records information about a skill required to complete a workflow task. These skill descriptions are used in conjunction with user skill descriptions to determine whether a user can perform a task on from a workqueue.

3.193.2 Properties

The table describes the object properties.

Table 3-187: Wq skill info object type properties

Property	Datatype	Single or repeating	Description
display_text	string(255)	R	Text displayed for the skill value at the corresponding index position in one of the valid_values_datatype properties.
multiple_select	integer	S	Indicates whether a user can have multiple values for this skill. Valid values are: <ul style="list-style-type: none"> • 0, the user may not have multiple values for this skill (default) • 1, the user may have multiple values for this skill
skill_description	string(2000)	S	Description of the skill.
skill_name	string(128)	S	Name of the skill.

Property	Datatype	Single or repeating	Description
skill_operator	integer	S	<p>Operator to use when checking a user's actual skill level meets the required skill for a workflow task. Valid values depend on the datatype of the skill.</p> <p>For skills of type integer, valid values are:</p> <ul style="list-style-type: none"> • 0, meaning != • 1, meaning = • 2, meaning > • 3, meaning >= • 4, meaning < • 5, meaning <= • 6, meaning <i>bit and</i> • 7, meaning <i>bit or</i> <p>For skills of type double, valid values are:</p> <ul style="list-style-type: none"> • 0, meaning != • 1, meaning = • 2, meaning > • 3, meaning >= • 4, meaning < • 5, meaning <= <p>For skills of type string, valid values are:</p> <ul style="list-style-type: none"> • 0, meaning <i>not equals</i> • 1, meaning <i>equals</i>
skill_type	integer	S	<p>Datatype of the skill. Valid values are:</p> <ul style="list-style-type: none"> • 0, integer (default) • 1, double • 2, string

Property	Datatype	Single or repeating	Description
valid_values_double	double	R	Valid values of the skill. This property is used only when the skill_type is 1.
valid_values_int	integer	R	Valid values of the skill. This property is used only when the skill_type is 0.
valid_values_string	string(255)	R	Valid values of the skill. This property is used only when the skill_type is 2.

3.194 Wq task skill

Associates a required skill with a workflow task.

3.194.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmc_wq_task_skill
- Object type tag: 00

A wq task skill object associates a skill with a workflow task on a workqueue.

3.194.2 Properties

The table describes the object properties.

Table 3-188: Wq task skill object type properties

Property	Datatype	Single or repeating	Description
skill_info_id	ID	S	Object ID of the dmc_wq_skill_info object.
skill_values_double	double	R	The required skill values for the task. This is set only when the skill is a double datatype skill.

Property	Datatype	Single or repeating	Description
skill_values_int	integer	R	The required skill values for the task. This is set only when the skill is a integer datatype skill.
skill_values_string	string(255)	R	The required skill values for the task. This is set only when the skill is a string datatype skill.
workitem_id	ID	S	Object ID of the work item.

3.195 Wq user skill

Records the skill values for a user for a specific skill.

3.195.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmc_wq_user_skill
- Object type tag: 00

A wq user skill object records the skill values for a particular user for a specific skill.

3.195.2 Properties

The table describes the object properties.

Table 3-189: Wq user skill object type properties

Property	Datatype	Single or repeating	Description
skill_info_id	ID	S	Object ID of the dmc_wq_skill_info object.
skill_values_double	double	R	The required skill values for the task. This is set only when the skill is a double datatype skill.

Property	Datatype	Single or repeating	Description
skill_values_int	integer	R	The required skill values for the task. This is set only when the skill is a integer datatype skill.
skill_values_string	string(255)	R	The required skill values for the task. This is set only when the skill is a string datatype skill.
user_name	string(255)	S	Name of the user.

3.196 XML application

Stores the information that defines a particular kind of XML document.

3.196.1 Description

- Supertype: Folder
- Subtypes: None
- Internal name: dm_xml_application
- Object type tag: 0b

The information in an xml application object includes the document type definition (DTD) used by the document, the name space identified in the document's header, and the document's root element.

3.196.2 Properties

The table describes the object properties.

Table 3-190: XML application type properties

Property	Datatype	Single or repeating	Description
dtd_public_id	string(255)	S	The public ID of the document type definition identified in dtd_system_id.
dtd_system_id	string(255)	S	The system ID of a document type definition.
namespace	string(80)	S	The name space identified in the XML document's header.

Property	Datatype	Single or repeating	Description
root_elements	string(32)	R	Specifies the type of XML document.
root_object_types	string(32)	R	Names of the object types used by the XML application. These are the formal names of the object types; for example, dm_document. Custom object types can be included. Documentum Desktop uses this property to determine which object types to display in user interface fields for operations on objects handled by an XML application.

3.197 XML config

Stores a document type definition file (DTD) as content.

3.197.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dm_xml_config
- Object type tag: 09

An xml config object stores, as content, a document type definition file (DTD), the XML file that describes the rules for chunking an XML document in an XML application.

3.197.2 Properties

The table describes the object properties.

Table 3-191: XML config type properties

Property	Datatype	Single or repeating	Description
config_locator	ID	S	Used internally to manage XML documents.

3.198 XML custom code

Stores the Java code that implements a custom Java interface as content.

3.198.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dm_xml_custom_code
- Object type tag: 09

An xml custom code object stores, as content, the Java code that implements a custom Java interface.

3.198.2 Properties

The table describes the object properties.

Table 3-192: XML custom code type properties

Property	Datatype	Single or repeating	Description
class_path	string(255)	S	Java class path of the code.

Property	Datatype	Single or repeating	Description
code_type	integer	S	Specifies the Java interface implemented by the code. Valid values are: <ul style="list-style-type: none">• 1, meaning code is a DfDTDHandler subclass• 2, meaning code is a DfDTDDocument Handler subclass• 3, meaning Code implements IDfLinkDetector
com_class_id	string(80)	S	Currently unused.

3.199 XML style sheet

Stores an XSL file as content.

3.199.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dm_xml_style_sheet
- Object type tag: 09

An xml style sheet object stores an XSL file as content. An XSL file is a style sheet that defines how to format an XML document on output.

3.199.2 Properties

The table describes the object properties.

Table 3-193: XML style sheet type properties

Property	Datatype	Single or repeating	Description
transformed _format	string(80)	S	Defines the output file format of the XML document.

3.200 XML zone

Contains the information used to populate the Zone tab on a Find dialog.

3.200.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dm_xml_zone
- Object type tag: 09

An xml zone object contains the information used to populate the Zone tab on a Find dialog.

3.200.2 Properties

The table describes the object properties.

Table 3-194: XML zone type properties

Property	Datatype	Single or repeating	Description
Description	string(32)	S	User defined.
zone_name	string(32)	R	Specifies one or more zones that can be searched in a fulltext index search. A zone is the name of an XML tag.

Chapter 4

Lightweight object types and shareable object type properties

4.1 Lightweight and shareable objects

When first introduced, the lightweight types and the properties supporting shareable types were only used by OpenText Documentum CM client products. Users could not create lightweight or shareable object types. From version 6.5, users can take advantage of lightweight and shareable types. There are additional properties defined for dm_type and dmi_type_info that also support these object types. Refer to the description of dm_type and dmi_type_info object types for more information about those properties.

Starting with release 6.0, non-qualifiable properties are supported for external use. When creating an object type, one or more of the type properties can be non-qualifiable. For more information about non-qualifiable properties and the supporting property bag property, see *OpenText Documentum Content Management - Server Fundamentals Guide (EDCCS250400-GGD)*. For more information about creating a type with a non-qualifiable property, see *OpenText Documentum Content Management - Server DQL Reference Guide (EDCCS250400-DRD)*.

4.2 Lightweight

4.2.1 Description

- Supertype: dm_sysobject
- Subtypes: User-defined
- Internal name: N/A
- Object type tag: inherited from supertype (or, for OpenText Documentum CM client types released in 6.0, 6E)

For OpenText Documentum CM client types this object type serves as the supertype of any lightweight object type. Similar to the persistent object type, it serves only to pass on its properties to all of its subtypes. For version 6.5, the data model was changed so that lightweight types are subtypes of SysObjects.

4.2.2 Properties

The table describes the object properties.

Table 4-1: Lightweight object type properties

Property	Datatype	Single or repeating	Description
i_vstamp	integer	S	Version stamp of the object.
i_sharing_parent	ID	S	Object ID of the parent object with which the lightweight object is sharing. If the object is materialized, this value is the object ID of the lightweight object.
object_name	string(255)	S	Name of the object.
r_object_id	ID	S	Object ID of the object.
r_page_cnt	integer	S	Number of primary content files associated with the object.

4.3 Lightweight, shareable, and non-qualifiable properties

Provide support for lightweight subtypes and shareable types and non-qualifiable properties.

4.3.1 Description

The properties described in the table are added to an object type only under specific conditions. The description of each property describes the conditions under the property appears in an object type definition.

Table 4-2: Lightweight subtypes, shareable subtypes, and non-qualifiable properties

Property	Datatype	Single or repeating	Description
i_sharing_type	ID	S	<p>This property is added to any object type defined as a shareable object type. The property records the object ID of the dm_type object representing the top-most lightweight object type in the type hierarchy of the lightweight object instance that shares the shareable instance.</p> <p>The property is NULL if no lightweight objects share the shareable instance.</p>
i_orig_parent	ID	S	<p>This property is added to any object type defined as a shareable object type. The property records the object ID of the shareable instance of the sharing, parent object.</p> <p>Consequently, for a materialized lightweight object, this value records the original sharing instance with which the lightweight object shared values.</p>
allow_propagating_changes	Boolean	S	<p>Indicates whether any changes applied to the sharing instance are propagated to the lightweight objects that share the instance.</p> <p>The default is F.</p>

Property	Datatype	Single or repeating	Description
i_property_bag	string(2000)	S	<p>The property stores the names and values of non-qualifiable properties. It is also used to store the names and values of aspect properties if the properties are added to the aspect with the OPTIMIZEFETCH option.</p> <p>This property is automatically added to the definition of a lightweight object type if the type definition contains a non-qualifiable property. The property can also be explicitly added by altering the object type.</p>
r_property_bag	string(2000)	R	<p>This property stores overflow from the i_property_bag property.</p> <p>This property is added to an object type definition if the definition includes the i_property_bag property.</p>

Chapter 5

Non-persistent object reference

5.1 Non-persistent object types

Non-persistent object types represent objects that are created as needed at runtime and are destroyed when the session is terminated.

5.2 Client config

An client config object describes the configuration parameters for sessions.

5.2.1 Description

The client config object type is a non-persistent type.

A client config object is created when Foundation Java API is initialized. Its properties are the keys of the dfc.properties file. If a key is set in the file, its value is reflected in the client config object. If a key is not specifically set, the client config object reflects the key's default value. The values are applied to all sessions opened by that Foundation Java API. This object is intended for use by system administrators.

5.2.2 Properties

The object and the object property values can be accessed using the IDfSession.getClientConfig method.

5.3 Connection config

Describes a session's connection to a single repository.

5.3.1 Description

The connection config object type is a non-persistent type.

A connection config object describes a session's connection to a single repository. This object is intended for use by system administrators.

5.3.2 Properties

The table describes the object properties.

Table 5-1: Connection config type properties

Property	Datatype	Single or repeating	Description
client_cache_size	integer	S	Defines the size of the client cache size for the subconnection. The value is taken from the session config's client_cache_size property setting.
client_cache_write_interval	integer	S	Controls how often periodic refreshes of the client persistent cache occurs. The default value is 60 minutes.
connection_id	ID	S	Object ID of the session object for the connection.
connection_name	string(5)	S	Connection identifier for the connection.
force_coherency_checks	Boolean	S	T disables the use of consistency check rules (for client persistent caches) defined in queries or a cache config object. The default is F (FALSE).
network_requests	integer	S	Number of RPC calls sent by the client session to the server. This updates throughout the session, whenever an RPC call occurs.
r_date_format	string(40)	S	The date format that will be used to return dates to the user.
r_docbase_id	ID	S	repository ID of the repository to which the subconnection is connected.

Property	Datatype	Single or repeating	Description
r_docbase_name	string(120)	S	Name of the repository to which the subconnection is connected.
r_events_location	string(32)	S	Name of the location object in the repository that points to the event directory of the repository.
r_mac_protocol	string(32)	S	Specifies the Macintosh file-sharing protocol used by the repository.
r_persistent_caching	Boolean	S	Specifies whether persistent client caching is enabled for the session. T (TRUE) indicates that caching is enabled. F (FALSE) indicates that it is disabled.
r_security_mode	string(32)	S	Specifies the security mode under which the repository is running.
r_user_name	string(255)	S	Name of the current user. This value is always the same as that found in the session config object.
secure_channel	Boolean	S	Specifies whether the session is using a secure connection for its repository connection. T (TRUE) means the connection is secure. F (FALSE) means it is not.

5.4 Docbase locator

Contains information about the repositories known to a connection broker.

5.4.1 Description

The docbase locator type is a non-persistent type.

A docbase locator object is an object that is constructed and returned by a connection broker in response to a Getdocbasemap method call. The information for a single repository appears at corresponding index levels in the repeating properties. For example, the name of the repository whose ID appears in r_docbase_id[0] is found in r_docbase_name[0], and its description is found in r_docbase_description[0].

5.4.2 Properties

The table describes the object properties.

Table 5-2: Docbase locator type properties

Property	Datatype	Single or repeating	Description
auth_protocol	string(32)	S	On Windows platforms, if set to domain required, indicates that the repository is running in domain-required mode. If the repository is not using domain-required mode, this property is blank. On Linux platforms, if you are authenticating users against a Windows domain, this is set to unix_domain_used. Otherwise, this property is blank for repositories running on a Linux platform.
i_docbroker_version	string(32)	S	Version number of the responding connection broker.

Property	Datatype	Single or repeating	Description
i_host_addr	string(32)	S	The IP address of the host on which the responding connection broker resides.
i_host_name	string(128)	S	Name of the host machine on which the responding connection broker resides.
i_port_number	integer	S	Port number of the responding connection broker.
r_docbase_description	string(128)	R	Verbose name or description of the repository.
r_docbase_id	ID	R	The internal ID of the repository.
r_docbase_name	string(32)	R	Name of the repository.
r_federation_name	string(32)	R	Name of the federation to which this docbase belongs, if any.
r_govern_docbase	string(32)	R	For each repository participating in a federation, the name of the governing repository in the federation.
r_object_id	ID	S	Object ID of the docbase locator object.
r_server_version	string(32)	R	Version numbers of the servers.

5.5 Docbroker locator

Contains information about each connection broker that the client DMCL can access.

5.5.1 Description

The docbroker locator type is a non-persistent type.

A docbroker locator object contains information about each connection broker that the client DMCL can access. The object is constructed and returned by the client library in response to a Getdocbrokermap method call. The information for a single connection broker appears at corresponding index levels in the properties. For example, the protocol for the connection broker whose host_name appears in host_name[0] is found in network_protocol[0] and its port number is found in port_number[0].

5.5.2 Properties

The table describes the object properties.

Table 5-3: Docbroker locator type properties

Property	Datatype	Single or repeating	Description
host_name	stringstring((128))	R	Name of the host machine on which the connection broker resides.
network_protocol	string(12)	R	Protocol for a given connection broker.
port_number	integer	R	Number of the port on the host machine that the connection broker uses for communication.
time_out	integer	R	Time, in seconds, that a server waits for a response from the connection broker before forwarding the request to a backup connection broker.

5.6 Server locator

Contains information about the servers known to a connection broker.

5.6.1 Description

The server locator type is a non-persistent type

A server locator object contains information about the servers known to a connection broker. A server locator object is constructed and returned by the connection broker in response to a Getservermap method call.

5.6.2 Properties

The table describes the object properties.

Table 5-4: Server locator type properties

Property	Datatype	Single or repeating	Description
a_silent_login	Boolean	S	Used internally by the OpenText Documentum CM clients.
i_connection_protocol	string(12)	R	Used internally by the dmcl to establish a network connection.
i_docbase_id	string	S	Decimal value of the DocBase identifier assigned during installation.
i_docbroker_version	string(32)	S	Version number of the responding connection broker.
i_host_addr	string(32)	S	IP address of the host machine on which the responding connection broker resides.
i_host_name	string(128)	S	Name of the host machine on which the responding connection broker resides.

Property	Datatype	Single or repeating	Description
i_port_number	integer	S	Port number of the port on the host machine that the connection broker is using for communications.
i_server_connection_address	string(40)	R	Used internally by the dmcl to establish a network connection.
i_logon_support	string(64)	S	Type of logon (it is NT_Unified_logon for servers running under Windows).
r_client_proximity	integer	R	Indicates how far the server is from the client. The value rises directly in relation to the distance.
r_host_name	string(128)	R	Name of the host machine on which the server resides.
r_keep_entry_interval	integer	R	How long the connection broker will keep the server's entry in the absence of a checkpoint message from the server.
r_last_checkpoint	TIME	R	The time that the server last reported to the connection broker.
r_last_status	string(24)	R	Status of the server process. The value is one of: starting, open, stopped, or presumed down.
r_next_checkpoint	TIME	R	The time that the server is next expected to report to the connection broker.
r_object_id	string	S	Object ID of the server locator object.

Property	Datatype	Single or repeating	Description
r_process_id	integer	R	Process ID of the server.
r_server_name	string(128)	R	Name of the server.
r_server_version	string(32)	R	Version number of the server.

5.7 Session config

Contains information about an open repository session.

5.7.1 Description

The session config object type is a non-persistent type.

A session config object contains information about an open repository session. Each repository session has one associated session config object. Access to this object is through the IDfSessionManager interface.

5.7.2 Properties

The table describes the object properties.

Table 5-5: Session config type properties

Property	Datatype	Single or repeating	Description
alias_set	string(32)	S	The session-level default alias set.
api_exec_count	integer	S	Number of dmAPIExec function calls issued during the session.
api_get_count	integer	S	Number of dmAPIGet function calls issued during the session.
api_set_count	integer	S	Number of dmAPISet function calls issued during the session.

Property	Datatype	Single or repeating	Description
application_code	string(32)	R	<p>Specifies, by application, the application-controlled objects that this session can modify. If this is NULL, the session cannot modify any application-controlled objects.</p> <p>Application codes can contain only alphanumeric characters and the underscore character. They cannot contain spaces nor can they start with the characters dm_. Codes beginning with dm_ are reserved for use by OpenText Documentum CM.</p>
batch_hint_size	integer	S	Defines the number of rows returned to Documentum CM Server by the RDBMS in a single call to the RDBMS. The default is 20.
client_cache_size	integer	S	Obsolete.
docbase_scope	string(120)	S	Obsolete.
dynamic_groups	string(32)	R	List of the dynamic groups to which the session belongs.
ignore_pre_processing	Boolean	S	Used for customized workflows. It directs the server not to execute external applications defined as pre-processing. The default is FALSE.

Property	Datatype	Single or repeating	Description
ignore_post_processing	Boolean	S	Used for customized workflows. It directs the server not to execute external applications defined as post-processing. The default is FALSE.
local_clean_on_init	Boolean	S	Indicates whether the server should automatically purge the client local area whenever the client is started. The default is T.
local_diskfull_check	integer	S	Defines how often the server checks the free space on the disk containing the client's local area. The default is 1, meaning that the space is checked with each getFile. Setting this to a number greater than 1, n, means that the space is checked every n getFiles. Setting it to 0 means the space is never checked.
local_diskfull_limit	integer	S	Specifies the maximum amount of space that the client local area may use. The value is interpreted in megabytes. A value of 0 means no limit.
local_purge_on_diskfull	Boolean	S	Indicates whether the server should automatically purge the client local area when the diskfull limit is reached. The default value is T.
local_path	string(255)	S	Specifies the location of the local client area. Use a full path specification.

Property	Datatype	Single or repeating	Description
network_requests	integer	S	Number of RPC calls made during the session. This is a cumulative number, incremental by one each time an RPC call is made in the session.
r_cache_query	Boolean	S	Indicates whether the session is enabled for query caching. The default value is FALSE.
r_date_format	string(40)	S	The date format the server will use to return dates to the client. The value is derived from the setting in the connectionconfig object for the first repository connection of the session.
r_events_location	string(32)	S	Specifies the location object in the repository representing the events directory. The value is derived from the setting in the connection config object for the first repository connection of the session.
r_host_name	string(128)	S	Contains the host name of the machine on which the current session was initiated.
r_mac_protocol	string(32)	S	Specifies the file-sharing protocol if this is a Macintosh client session. The value is derived from the setting in the connection config object for the first repository connection of the session.

Property	Datatype	Single or repeating	Description
r_security_mode	string(32)	S	Contains the security level enforced for the repository you are using. The value is derived from the setting in the connection config object for the first repository connection of the session.
r_user_name	string(32)	S	Contains the user name of the currently logged-in user.
r_working_directory	string(255)	S	Contains the path of the user's working directory.
ref_binding_label	string(32)	S	Defines which document version to fetch when accessing a remote document. If this property is defined, its value overrides the binding specified in the dm_reference object that points to the remote object.
session_alias	string(12)	S	Contains the session identifier, as a string literal.

Property	Datatype	Single or repeating	Description
session_codepage	string(64)	S	<p>The code page in use for the repository session. By default, the value is obtained from the client_codepage property of the api config object.</p> <p>If client_codepage is unset, the value is determined programmatically based on the value in session_locale.</p> <p>For all session locales except ja (Japanese) and ko (Korean), the default setting is ISO-8859-1.</p> <p>For Japanese, the default setting is Shift_JIS.</p> <p>For Korean, the default setting is EUC-KR.</p>
session_id	ID	S	Contains the ID of the session.

Property	Datatype	Single or repeating	Description
session_locale	string(32)	S	<p>Locale of the repository session. By default, the value is obtained from the client_locale property of the api config object. If client_locale is unset, the setting is determined programmatically based on the locale of the client's host machine.</p> <p>Valid values are:</p> <ul style="list-style-type: none">• en, for English• de, for German• fr, for French• ja, for Japanese• ko, for Korean

Chapter 6

Content Aviator object reference

6.1 dm csai config

The table describes the Content Aviator object properties.

Table 6-1: Schema properties

Property	Datatype	Single or repeating	Description
llm_enabled	Boolean	S	Used to enable or disable the CSAI feature during deployment. The default value is true.
llm_modified_date	date	S	Specifies the modified date.
llm_modified_user	char(255)	S	Specifies the modified user name details.
embd_svc_url	char(255)	S	CSAI embedding service URL. For example: http://<CSAI>/embeddings.
query_svc_url	char(255)	S	CSAI chat or query service URL. For example: http://<CSAI>/chat
activemq_url	char(255)	S	Specifies the ActiveMQ URL.

Property	Datatype	Single or repeating	Description
activemq_user	char(255)	S	ActiveMQ user name created for embedding queue only. This user is created during the AMQ deployment and is restricted to create or read or delete message for the embedding queue. For example: artemis
activemq_password	char(255)	S	Password for the embedding queue user.
activemq_qname	char(255)	S	Specifies the ActiveMQ name.
client_id	char(255)	S	Specifies the client ID details.
client_secret	char(255)	S	Specifies the client ID secret details.
activemq_folder_qname	char(255)	S	Specifies the folder queue name.
extend_params	char(255)	R	Reserved for future use.

6.2 dcis document aspect

The table describes the dcis document aspect object properties.

Table 6-2: Document schema properties

Property	Datatype	Single or repeating	Description
enabled_user	char(255)	S	Specifies the LLM enabled user.
embed_enabled_date	date	S	Specifies the LLM enabled date.
embed_mod_date	date	S	Specifies the LLM embedding status modified date.
embed_status	char(100)	S	Specifies the LLM embedding status.

Property	Datatype	Single or repeating	Description
embed_remark	char(1024)	S	Reserved for future use.
i_partition	integer	S	Reserved for future use.
ancestries	char(255)	R	Contains the folder IDs with enabled with Aviator, including linked folders.
extra_params	char(255)	R	Reserved for future use.
is_explicitly_enabled	Boolean	S	The default value is False. It changes to True when an Admin user enables the virtual document for aviator folder.
enabled_from_vd	Boolean	R	When you select 0, the virtual document or general document is enabled by using folder enablement. When you select 1, the virtual document is enabled by using a virtual document enablement.
enqueue_sequence	int	S	To manage the processing order, DCIS engine uses this property for embedding.

6.3 dcis folder aspect

The table describes the dcis folder aspect object properties.

Table 6-3: Folder schema properties

Property	Datatype	Single or repeating	Description
embed_enabled	Boolean	S	Specify the LLM enabled status for folder.
embed_enabled_date	date	S	Specify the LLM enabled date.

Property	Datatype	Single or repeating	Description
enabled_user	char(255)	S	Specify the LLM enabled user.
criteria	char(255)	S	Reserved for future use.
extra_params	char(128)	S	Reserved for future use.
embd_status	char(100)	S	Specify the LLM embedding status.
i_partition	integer	S	Reserved for future use.
ancestries	char(255)	R	Contains the folder IDs enabled with Aviator, including linked folders.
extra_params	char(255)	R	Reserved for future use.
is_explicitly_enabled	Boolean	S	By default, this value is set to False. It changes to True when the Admin user enables the folder for aviator .
enqueue_sequence	integer	S	To manage the processing order, DCIS engine uses this property for embedding.

Chapter 7

Collaboration object reference

7.1 Collaboration object types

Collaboration object types support Documentum Collaborative Services. The object types are packaged in the Documentum Collaborative Services DAR file. Instances of the types are created and managed internally in conjunction with Documentum Collaborative Services.

7.2 Calendar

Serves as the base type for calendars.

7.2.1 Description

- Supertype: Folder
- Subtypes: None
- Internal name: dmc_calendar
- Object type tag: 0b

A calendar object contains the calendar event objects that comprise the calendar. Any objects in the calendar folder that are not calendar events are considered attachments to the calendar. These objects are created and managed internally when users create calendars.

7.2.2 Properties

The calendar object type inherits all properties from its folder supertype.

7.3 Calendar event

Records an individual event on a calendar.

7.3.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_calendar_event
- Object type tag: 08

A calendar event object stores the information about an event on a calendar. These objects are created and managed internally when users create and modify calendars. Calendar event objects are stored in a calendar folder representing the calendar on which the event appears.

7.3.2 Properties

The table describes the object properties.

Table 7-1: Calendar event type properties

Property	Datatype	Single or repeating	Description
change_number	integer	S	A value that is incremented whenever a change is made that invalidates attendee responses.
end_date	date	S	Event's end date.
event_attendees	ID	R	Object ID of the dm_user objects of the event participants who have user objects in the repository.
event_description	string This is Non-qualifiable property, so it has no defined length.	S	Rich text description of the event.
event_location	string(128)	S	Location of the event.
event_organizer	ID	S	Object ID of the repository user who organized the event.

Property	Datatype	Single or repeating	Description
event_priority	integer	S	Represents the priority of the event. Valid values are from 1 (highest priority) through 9 (lowest priority).
event_uid	string(255)	S	Globally unique identifier for the event.
exception_dates	date	R	For recurring events, the date of an exception to the recurrence rule. These are the dates on which the event did not occur as scheduled.
has_date_only	Boolean	S	T means that the start and end dates of the event specify only the date, with no time component. F means that the start and end dates include a time component.
is_transparent	Boolean	S	Specifies whether the event appears to consume time on the calendar. T means that event does appear to consume time. F means that the event does not appear to consume time on the calendar.
other_attendees	string(255)	R	List of email addresses of those participants who are not users in the repository.
recurrence_dates	date	R	Dates of recurrence instances.
recurrence_end	date	S	The upper boundary of the recurrence pattern.

Property	Datatype	Single or repeating	Description
recurrence_id	date	S	Date of recurrence which this event overrides.
recurrence_rule	string This is Non-qualifiable property, so it has no defined length.	S	Recurrence rule The recurrence rule is specified in the iCal RRULE format.
start_date	date	S	The starting date of the event.

7.4 Comment

Records a comment in a discussion

7.4.1 Description

- Supertype: RichText
- Subtypes: None
- Internal name: dmc_comment
- Object type tag: 08

A comment object represents a single comment in discussion. Comments are created and managed through Webtop.

7.4.2 Properties

The table describes the object properties.

Table 7-2: Comment type properties

Property	Datatype	Single or repeating	Description
comment_creation_date	date	S	Original date and time at which a comment was created. This is only set if the comment object represents a comment copied from the original comment.

Property	Datatype	Single or repeating	Description
comment_creator	string (128)	S	Name of the user who originally made the comment. This is only set if the comment object represents a comment copied from the original comment.
comment_id	integer	S	Identifying value for the comment within the topic.
comment_modtag	integer	S	Value of the topic's last_update_modtag property after the most recent modification of this comment.
comment_parentid	integer	S	The identifying value of the comment to which this comment is a reply or response. If the comment is not a reply to another comment, this is 0.

7.5 Datatable

Contains the objects that comprise a data table.

7.5.1 Description

- Supertype: Folder
- Subtypes: None
- Internal name: dmc_datatable
- Object type tag: 0b

A datatable object contains all the row objects, the datatable schema object, and the folders for row attachments that together define a data table. The properties of the datatable record the basic metadata about the datatable, such as its name and owner. These objects are created internally when a user creates a database in Collaboration Services.

7.5.2 Properties

The table describes the object properties.

Table 7-3: Datatable type object properties

Property	Datatype	Single or repeating	Description
schema_id	ID	S	Object ID of the dmc_datatable_schema object associated with this datatable.

7.6 Datatable row

Serves as the supertype for the custom types created to record the definition of a row in a datatable.

7.6.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_datatable_row
- Object type tag: 08

The datatable row object type is the supertype of all custom object types created to represent rows in a datatable. These custom types are created when a user creates a datatable. Each row has a custom object type that records the row's characteristics.

7.6.2 Properties

There are no properties defined for this object type. It inherits all properties from its supertype.

7.7 Datatable schema

Records metadata about a datatable.

7.7.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_datatable_schema
- Object type tag: 08

A datatable schema object records metadata about a datatable in its defined properties and stores the XML schema that contains information about the table fields as content. These objects are created and managed internally, as users create, modify, and delete datatables.

7.7.2 Properties

The table lists the properties defined for the type.

Table 7-4: Datatable schema type properties

Property	Datatype	Single or repeating	Description
column_counter	integer	S	Counter that provides a unique suffix to the row type attribute name, in the format: column_x, where x is an integer.
is_enterprise	Boolean	S	Indicates whether a schema can be shared among multiple instances of a data table.
row_type	string(32)	S	Name of the Documentum CM Server type used for the row.
title_column	string(32)	S	Identifier of the column which represents the name of the row.

7.8 Datatable schema ex

Records information required to manage datatable schemas.

7.8.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmc_datatable_schema_ex
- Object type tag: 00

A datatable schema ex object is used internally to manage datatables. Do not modify, delete, or change this type or one of these objects directly.

7.8.2 Properties

The table lists the properties defined for the type.

Table 7-5: Datatable schema ex type properties

Property	Datatype	Single or repeating	Description
a_column_id	string(32)	S	Specifies the column to which the data belongs. The value is the column_id.
a_next_autonumber	integer	S	Next automatically generated number to use for an auto-number field.
a_schema_id	ID	S	Object ID of the schema.

7.9 Datatable settings

7.9.1 Description

- Supertype: Relation
- Subtypes: None
- Internal name: dmc_datatable_settings
- Object type tag: 37

A datatable settings object links a datatable and its datatable schema object or a datatable and a user object. The properties of the datatable settings object store basic information about the settings of the datatable.

7.9.2 Properties

The table lists the properties defined for the type.

Table 7-6: Datatable settings type properties

Property	Datatype	Single or repeating	Description
fulltext_search_data	string(255)	S	Full-text search string data value.
grouping_column	string(32)	S	ID of the group column.
grouping_columns	string(32)	R	IDs of the columns that can be grouped.
search_column	string(32)	R	List of the columns that are searchable.
search_column_data	string(255)	R	Search parameters or data used to build a search query.
search_column_data2	string(255)	R	Additional search parameters or data used to build a search query.
search_column_opcode	integer	R	Specifies the operation code used to build the query.
sort_column	string(32)	S	Search column identifier.
summary_column	string(32)	R	List of summary columns.

7.10 Discussion

Contains one or more topics or topic subtypes.

7.10.1 Description

- Supertype: Folder
- Subtypes: None
- Internal name: dmc_discussion
- Object type tag: 0b

A discussion object serves as the container for topics relating to a particular discussion. Only objects of type dm_topic or its subtypes may be linked to a discussion.

7.10.2 Properties

The discussion object type has no properties defined for the type. It inherits all its properties from its supertype.

7.11 Notepage

Represents rich text content that is authored and read online.

7.11.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dmc_notepage
- Object type tag: 09

A notepage object stores a content file in rich text format. You cannot create renditions of notepage objects. The content represented by notepage objects is created and used in the context of collaboration in Webtop.

7.11.2 Properties

The table lists the property defined for the type.

Table 7-7: Notepage type properties

Property	Datatype	Single or repeating	Description
anchor_id	ID	R	Object IDs of any repository objects within the content file associated with the object.

7.12 Readcomment

Used to manage unread comments in a discussion.

7.12.1 Description

- Supertype: Persistent Object
- Subtypes: None
- Internal name: dmc_readcomment
- Object type tag: 00

A readcomment object contains information that allows the system to determine which comments in a particular discussion are unread by a particular viewer.

7.12.2 Properties

The table lists the properties defined for this type.

Table 7-8: Readcomment type properties

Property	Datatype	Single or repeating	Description
topic_id	ID	S	Object ID of topic object.
topic_viewed_modtag	integer	S	Date and time at which the user in user_name last viewed the discussion represented by the topic. The value is copied from the topic object's last_update_modtag property.
user_name	string(32)	S	Name of the user.

7.13 Richtext

Represents rich text content associated with a SysObject.

7.13.1 Description

- Supertype: SysObject
- Subtypes: dmc_comment
- Internal name: dmc_richtext
- Object type tag: 08

A richtext object represents richtext content associated with a SysObject either directly, through a relationship, or indirectly as a comment in a topic thread. The actual content is stored in either the content_value property or as an actual content file associated with the richtext object.

7.13.2 Properties

The table lists the properties defined for the type.

Table 7-9: Richtext type properties

Property	Datatype	Single or repeating	Description
anchor_id	ID	R	Object IDs of repository objects found in the content.
content_value	string(2000)	S	Stores the first 2000 bytes of the content.
has_content	Boolean	S	Specifies whether the content is fully contained in the content_value property or whether there is also a content file attached to the object. T means that there is a content file. F means that the content is fully contained in content_value.

7.14 Room

Provides an additional access management layer for SysObjects.

7.14.1 Description

- Supertype: Folder
- Subtypes: None
- Internal name: dmc_room
- Object type tag: 0b

A room is a special folder that provides additional, optional functionality to control access to the objects in the folder when the objects are accessed through the room.

7.14.2 Properties

The table lists the properties defined for the type.

Table 7-10: Room type properties

Property	Datatype	Single or repeating	Description
builtin_groups	string(32)	R	<p>Names of private groups in the room.</p> <p>The first four index positions are reserved for the system-defined groups in room:</p> <ul style="list-style-type: none"> • [0]=Members • [1]=Owners • [2]=Contributors • [3]=Visitors
default_contrib_perm_it	integer	S	Default base object-level permission for the contributor's group applied to objects governed by this room.
default_contrib_xperm_it	string(32)	S	Default extended permission for the contributor's group applied to objects governed by this room.

Property	Datatype	Single or repeating	Description
default_owner_permission	integer	S	Default base object-level permission for the owner's group applied to objects governed by this room.
default_owner_xperm	string(32)	S	Default extended permission for the owner's group applied to objects governed by this room.
is_public	Boolean	S	<p>Specifies whether the room is a public or private room.</p> <p>T means the room is a public room.</p> <p>F means the room is a private room and only members of the Members group (<code>builtin_groups[0]</code>) can access objects in the room.</p> <p>The only valid value is currently F.</p>

Property	Datatype	Single or repeating	Description
only_owners_ungovern	Boolean	S	<p>Specifies whether membership in the Owners groups is an additional requirement for users trying to remove an object from the room.</p> <p>T means that a user must be a member of the Owners group in addition to having Write and Change_permit permissions on an object to remove that object from the room.</p> <p>F means that any user in the room with Write and Change_permit permissions on the object can remove it from the room.</p>
status_text	string (<i>maximum</i>)	S	<p>Text description of the room's status.</p> <p>The length of this property is the maximum allowed by the underlying database.</p>
status_value	integer	S	Application-defined status value of the room.

7.15 Topic

Used to manage a single discussion.

7.15.1 Description

- Supertype: Folder
- Subtypes: None
- Internal name: dmc_topic
- Object type tag: 0b

A topic object is used to manage a discussion thread—the set of comments about a single topic.

7.15.2 Properties

The table lists the properties defined for the type.

Table 7-11: Topic type properties

Property	Datatype	Single or repeating	Description
is_disabled	Boolean	S	Specifies whether the topic is disabled. T means the topic is disabled. F means the topic is not disabled. The default is F.  Note: Users cannot create, view or edit comments in disabled topics.
next_comment_id	integer	S	Value to be used as the object ID of the next new comment.
last_update_modtag	integer	S	Value used internally to track each time a topic is modified.

Chapter 8

Document Submission Manager object reference

8.1 Document Submission Manager object types

Document submission manager object types support Document Submissions Manager. The object types are included and installed with the DSM DAR file.



Caution

These object types are used by Documentum Submissions Manager. You must not modify the object types, nor can you create instances of the types directly.

8.2 DSM application

Contains a eCTD application.

8.2.1 Description

- Supertype: Folder
- Subtypes: None
- Internal name: dmc_dsm_application
- Object type tag: 0b

A dsm application object contains the objects that comprise an eCTD application submitted and managed by Documentum Submissions Manager.

8.2.2 Properties

The table describes the object properties.

Table 8-1: Dmc dsm application type properties

Property	Datatype	Single or repeating	Description
last_published_name	string(255)	S	Name of the last published submission.

8.3 DSM backbone

Represents the root document of the virtual document that represents a submission.

8.3.1 Description

- Supertype: dsm section
- Subtypes: DSM Stf Backbone, DSM M1 Backbone
- Internal name: dmc_dsm_backbone
- Object type tag: 09

A dsm backbone object stores the index.xml file for a submission. This file is the root document of the virtual document that represents the submission in the repository.

8.3.2 Properties

The table describes the object properties.

Table 8-2: Dmc dsm backbone type properties

Property	Datatype	Single or repeating	Description
document_id	ID	R	Used internally.
dsm_data_model_version	string(32)	S	Used internally.
is_import_post_processed	Boolean	S	Used internally.
is_published	Boolean	S	Used internally.
last_used_leaf_id	string(128)	S	Used internally.
leaf_id	string(128)	R	Used internally.

8.4 DSM doc properties

Stores the values of the eCTD properties for an eCTD content file.

8.4.1 Description

- Supertype: Relation
- Subtypes: None
- Internal name: dmc_dsm_doc_properties
- Object type tag: 37

A dsm doc properties object records the eCTD properties of an eCTD content file. The values are version-specific, so there is one dsm doc properties object for each version of the document that contains the file.

8.4.2 Properties

The table describes the object properties.

Table 8-3: Dmc dsm doc properties type properties

Property	Datatype	Single or repeating	Description
application_version	string(128)	S	Version of the software application used to create the file.
checksum_type	string(128)	S	Algorithm used to generate the checksum.
checksum_value	string(128)	S	Checksum value for the file.
document_keywords	string(4000)	S	Keywords specified for the file.
document_title	string(128)	S	Title of the document.
font_library	string(128)	S	Commercial name of the font or font set used to create the document.
last_checksum_time	Date	S	Timestamp of the last checksum for this document.
version_identifier	string(128)	S	Submitter's internal version number or version identification for the report.

8.5 DSM drug product

Records information about a drug product specified in an eCTD section element.

8.5.1 Description

- Supertype: DSM Section
- Subtypes: None
- Internal name: dmc_dsm_drug_product
- Object type tag: 09

A dsm drug product object records the property values specified in a <m2-3-drug-product> or <m3-2-p-drug-product> section element.

8.5.2 Properties

The table describes the object properties.

Table 8-4: Dmc dsm drug product type properties

Property	Datatype	Single or repeating	Description
dosage_form	string(128)	S	The form in which the drug was given to study subjects.
manufacturer_name	string(128)	S	Name of the drug's manufacturer.
product_name	string(128)	S	Name of the product.

8.6 DSM drug substance

Records information about a drug substance specified in an eCTD section element.

8.6.1 Description

- Supertype: DSM Section
- Subtypes: None
- Internal name: dmc_dsm_drug_substance
- Object type tag: 09

A dsm drug substance object records the property values specified in a <m2-3-s-drug-substance> or <m3-2-s-drug-substance> section element.

8.6.2 Properties

The table describes the object properties.

Table 8-5: Dmc dsm drug substance type properties

Property	Datatype	Single or repeating	Description
manufacturer_name	string(128)	S	Name of the drug's manufacturer.
substance_name	string(128)	S	Name of the substance.

8.7 DSM excipient

Records information about an excipient substance specified in an eCTD section element.

8.7.1 Description

- Supertype: DSM Section
- Subtypes: None
- Internal name: dmc_dsm_excipient
- Object type tag: 09

A dsm excipient object records the property values specified in a <m3-2-a-1-control-of-excipients> section element.

8.7.2 Properties

The table describes the object properties.

Table 8-6: Dmc dsm excipient type properties

Property	Datatype	Single or repeating	Description
excipient_substance	string(128)	S	Name of the excipient substance.

8.8 DSM facilities equip

Records the property values specified in an eCTD section element.

8.8.1 Description

- Supertype: DSM Section
- Subtypes: None
- Internal name: dmc_dsm_facilities_equip
- Object type tag: 09

A dsm facilities equip object records the property values in a <m3-2-a-1-facilities-and-equipment> eCTD section element.

8.8.2 Properties

The table describes the object properties.

Table 8-7: Dmc dsm facilities equip type properties

Property	Datatype	Single or repeating	Description
dosage_form	string(128)	S	The form in which the drug was given to study subjects.
manufacturer_name	string(128)	S	Name of the drug's manufacturer.
product_name	string(128)	S	Name of the product.
substance_name	string(128)	S	Name of the substance.

8.9 DSM indication

Records the property values specified in an eCTD section element.

8.9.1 Description

- Supertype: DSM Section
- Subtypes: None
- Internal name: dmc_dsm_indication
- Object type tag: 09

A dsm indication object records the property values specified in a <m5-3-5-reports-of-efficacy-and-safety-studies> or <m2-7-3-summary-of-clinical-efficacy> eCTD section element.

8.9.2 Properties

The table describes the object properties.

Table 8-8: Dmc dsm indication type properties

Property	Datatype	Single or repeating	Description
indication_name	string(128)	S	Name of the indication.

8.10 DSM M1 backbone

Represents as regional backbone.

8.10.1 Description

- Supertype: DSM Backbone
- Subtypes: None
- Internal name: dmc_dsm_m1_backbone
- Object type tag: 09

A dsm m1 backbone object represents a regional backbone file.

8.10.2 Properties

The dsm m1 backbone object inherits all its properties from its parent type.

8.11 DSM safety eval

Records the property values specified in an eCTD section element.

8.11.1 Description

- Supertype: DSM Section
- Subtypes: None
- Internal name: dmc_dsm_safety_eval
- Object type tag: 09

A dsm safety eval object records the property values in a <m3-2-a-2-adventitious-agents-safety-evaluation> eCTD section element.

8.11.2 Properties

The table describes the object properties.

Table 8-9: Dmc dsm safety eval type properties

Property	Datatype	Single or repeating	Description
dosage_form	string(128)	S	The form in which the drug was given to study subjects.
manufacturer_name	string(128)	S	Name of the drug's manufacturer.
product_name	string(128)	S	Name of the product.
substance_name	string(128)	S	Name of the substance.

8.12 DSM sect doc attributes

Stores the values of attributes specific to a section-document relationship.

8.12.1 Description

- Supertype: Relation
- Subtypes: None
- Internal name: dmc_dsm_sect_doc_attributes
- Object type tag: 37

A dsm sect doc attributes object stores the values of properties specific to a section-document relationship. The relation_name for objects of this type is always dmc_dsm_section_to_doc.

8.12.2 Properties

The table describes the object properties.

Table 8-10: Dmc dsm sect doc attributes type properties

Property	Datatype	Single or repeating	Description
folder_path	string(1024)	S	Repository location of the document.
leaf_id	string(128)	S	Unique identifier for this file in the XML instance.
modified_file	string(128)	S	Specifies the location of the document being modified. The value in this property points the the index.xml file and the leaf ID of the file being altered.
modified_file_id	string(32)	S	Object ID of the document specified in modified_file.

Property	Datatype	Single or repeating	Description
operation_name	string(32)	S	Specifies the operation to be performed on the file to be modified. Valid values are: new append replace delete
submission_id	string(32)	S	Object ID of the submission folder to which the document belongs.

8.13 DSM section

Records information used to display a submission in Virtual Document Manager.

8.13.1 Description

- Supertype: Document
- Subtypes: DSM Backbone, DSM Drug Product, DSM Drug Substance, DSM Excipient, DSM Facilities Equip, DSM Safety Eval, DSM Indication, DSM Section Extension, DSM Stf Section
- Internal name: dmc_dsm_section
- Object type tag: 09

A dsm section object contains information used by Documentum Submissions Manager to manage the display of the submission in the Virtual Document Manager.

8.13.2 Properties

The table, describes the object properties.

Table 8-11: Dmc dsm section type properties

Property	Datatype	Single or repeating	Description
ectd_profile	string(32)	S	Specifies the associated DSM configuration profile governing DSM processing. This property may be empty. If so, the value is derived from the first parent in the virtual document that contains a value for this property.
section_id	string(128)	S	Name of the section. If the section has properties, the property values are concatenated with the section name.
section_name	string(128)	S	The eCTD section name. The exact value depends on whether the section has properties.

8.14 DSM section extension

Records eCTD element extensions.

8.14.1 Description

- Supertype: DSM Section
- Subtypes: None
- Internal name: dmc_dsm_section_extension
- Object type tag: 08

A dsm section extension object records eCTD element extensions. Each pair at the same index position in the properties represents a nested level of <node-extension>.

8.14.2 Properties

The table describes the object properties.

Table 8-12: Dmc dsm section extension type properties

Property	Datatype	Single or repeating	Description
extension_id	string(128)	R	List of extension identifiers.
extension_title	string(128)	R	Titles of the extensions. The title at any particular index position is associated with the identifier in the corresponding position in extension_id.

8.15 DSM stf backbone

Represents the backbone of a study tagging (stf.xml) file.

8.15.1 Description

- Supertype: DSM Backbone
- Subtypes: None
- Internal name: dmc_dsm_stf_backbone
- Object type tag: 09

A dsm stf backbone object represents the root document of the virtual document that represents an stf.xml file in the repository. That virtual document is a component of the virtual document whose root is represented by a dsm backbone object.

8.15.2 Properties

The table describes the object properties. For the repeating properties, the values recorded at a particular index position represent one category name attribute element.

Table 8-13: Dmc dsm stf backbone type properties

Property	Datatype	Single or repeating	Description
category_name	string(128)	R	Category names identified in the <category name> properties in a <study-identifier> element.
category_info_type	string(32)	R	Information type identified in <category name> properties in a <study-identifier> element.
category_value	string(128)	R	The value specified in <category name> properties in a <study-identifier> element.
study_id	string(128)	S	Value of the <study-id> attribute in a <study-identifier> element.
study_title	string(128)	S	Value of the <title> attribute in a <study-identifier> element.

8.16 DSM stf section

Records the attributes of an eCTD content block within an STF document.

8.16.1 Description

- Supertype: DSM Section
- Subtypes: None
- Internal name: dmc_dsm_stf_section
- Object type tag: 09

A dsm stf section object stores the values of attributes of an eCTD content block from an stf.xml document.

8.16.2 Properties

The table describes the object type properties.

Table 8-14: Dmc dsm stf section type properties

Property	Datatype	Single or repeating	Description
block_title	string(128)	S	The title value of the content block.
property_info_type	string(32)	R	The information types specified in property attributes.
property_name	string(128)	R	The names specified in property attributes.
property_value	string(128)	R	The values specified in property attributes.
subject_name	string(128)	R	Name of the subject.
subject_info_type	string(32)	R	The information type for the subject.

8.17 DSM study attributes

Records the study-related attributes for an STF study report.

8.17.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_dsm_study_attributes
- Object type tag: 08

A dsm study attributes object records the values of the study attribute for an STF study report. The objects are related to the document that stores the report. The relation name is always dmc_dsm_doc_to_study_attri.

8.17.2 Properties

The table describes the object properties.

Table 8-15: Dmc dsm study attributes type properties

Property	Datatype	Single or repeating	Description
property_info_type	string(32)	R	The information types specified in property attributes.
property_name	string(128)	R	The names specified in n property attributes.
property_value	string(128)	R	The values specified in property attributes.
study_title	string(128)	S	The title of the study.
subject_name	string(128)	R	Names of the subjects.
subject_info_type	string(32)	R	The information types for the subjects.

8.18 DSM study report

Relates a study report within an STF to a logical content file in the main backbone.

8.18.1 Description

- Supertype: Relation
- Subtypes: None
- Internal name: dmc_dsm_study_report
- Object type tag: 37b

A dsm study report object establishes a relationship between a study report within an STF and a logical content file in the main backbone. The objects are related to the document that stores the report. The relation name is always dmc_dsm_doc_to_stf.

8.18.2 Properties

The table describes the object type properties.

Table 8-16: Dmc dsm study report type properties

Property	Datatype	Single or repeating	Description
study_report_status	string(32)	S	Only valid value is Delete, to indicate that the logical content has been deleted from the STF.

8.19 DSM submission

8.19.1 Description

- Supertype: Folder
- Subtypes: None
- Internal name: dmc_dsm_submission
- Object type tag: 0b

A dsm submission object contains all the documents related to a particular submission.

8.19.2 Properties

The table describes the object properties.

Table 8-17: Dmc dsm submission type properties

Property	Datatype	Single or repeating	Description
previous_name	string(32)	S	Previous submission name before “lock”.

Chapter 9

Retention Policy Services object reference

9.1 Retention Policy Services object types

Retention policy object types are used by Retention Policy Services.



Note: Retention policy object types are installed with the Retention Policy Services product. Instances of these types are created and maintained through the Retention Policy Services product. Neither the types nor instances of the types can be changed or modified by users or client applications other than Retention Policy Services.

9.2 Retention Policy Services action

Specifies the Java class used to execute an action.

9.2.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_rps_action
- Object type tag: 08

An Retention Policy Services action object identifies the class and, optionally the interface within the class, that is executed to perform an action.

9.2.2 Properties

The table describes the object properties.

Table 9-1: Retention Policy Services action type properties

Property	Datatype	Single or repeating	Description
action_type	integer	S	Specifies what kind of action is represented by the Retention Policy Services action object. The only valid value is 0, meaning a notification.

Property	Datatype	Single or repeating	Description
class	string(128)	S	Path to the Java class that executes the action.
interface	string(128)	S	Path to the Java class interface that executes the action.
object_name	string(255)	S	Name of the action.
subject	string(192)	S	Description of the action.

9.3 Retention Policy Services action rel

Relates an Retention Policy Services phase rel object to an Retention Policy Services action object.

9.3.1 Description

- Supertype: Relation
- Subtypes: None
- Internal name: dmc_rps_action_rel
- Object type tag: 37

An Retention Policy Services action rel object associates a particular Retention Policy Services phase with an action and identifies the execution rule for the action.

9.3.2 Properties

The table describes the object properties.

Table 9-2: Retention Policy Services action rel type properties

Property	Datatype	Single or repeating	Description
arg_object_id	ID	R	Used internally. Value dependent on the type of action identified in the associated Retention Policy Services action object.
child_id	ID	S	Object ID of the Retention Policy Services action object.

Property	Datatype	Single or repeating	Description
execution_rule_id	ID	S	Object ID of the execution rule that is the criterion for action.
object_name	string(255)	S	Name of the object.
parent_id	ID	S	Object ID of the Retention Policy Services phase rel object representing the phase associated with the action.

9.4 Retention Policy Services authority

Specifies the persons or agencies that are authorized to determine whether an object has fulfilled its phase conditions.

9.4.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_rps_authority
- Object type tag: 08

An Retention Policy Services authority object records information about the authorities authorized to determine whether an object has fulfilled its phase conditions. The Retention Policy Services authority objects are related to objects of type dmc_rps_phase_rel objects (representing phases in a retention policy lifecycle). The relationship is named dmc_rps_phase_authority_rel_type.

9.4.2 Properties

The table describes the object properties.

Table 9-3: Retention Policy Services authority type properties

Property	Datatype	Single or repeating	Description
authorizers_id	ID	R	Object IDs of the Retention Policy Services contact objects representing users or agencies.

Property	Datatype	Single or repeating	Description
is_authority_valid	Boolean	S	Specifies whether this authority is valid. T means that this authority can determine phase fulfillment; F means that it cannot be used to determine phase fulfillment.
object_name	string(255)	S	Name of the authorizing body or person. This property is required.
title	string(400)	S	User-defined description of the authority.

9.5 Retention Policy Services base date

Records which property in an object type is used to populate the retention_base_date property in the dmc_rps_retainer objects for instances of the type.

9.5.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_rps_base_date
- Object type tag: 08

An Retention Policy Services base date object records the property in an object type whose value is used to populate the retention_base_date property in the dmc_rps_retainer objects with which instances of object type are associated.

9.5.2 Properties

The table describes the object properties.

Table 9-4: Retention Policy Services base date type properties

Property	Datatype	Single or repeating	Description
object_type	string(32)	S	Name of the object type. The object type must be dm_sysobject or one of its subtypes. Use the type's internal name. This is a required property.
base_date_name	string(32)	S	Name of an property defined for or inherited by the object type identified in object_type. The property must be of date/time datatype. This is a required property.
title	string(32)	S	User-defined description of the property mapping.

9.6 Retention Policy Services child strategy

Records the name of the class called to apply an Retention Policy Services retainer object to the objects associated with a dm_retainer object.

9.6.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_rps_child_strategy
- Object type tag: 08

An Retention Policy Services child strategy object defines the Java class called to attach an Retention Policy Services retainer to objects that are stored in a container that is associated with an Retention Policy Services retainer object.

9.6.2 Properties

The table describes the object properties.

Table 9-5: Retention Policy Services child strategy type properties

Property	Datatype	Single or repeating	Description
class	string(128)	S	Path to the Java class that executes based on the selected child strategy. This is a required property.
interface	string(128)	S	Path the interface for the class specified in class.
is_container_aging	Boolean	S	Specifies whether the Retention Policy Services retainer ages. T means the container ages; F means it does not.
object_name	string(255)	S	Name of the child strategy. Valid values are: <ul style="list-style-type: none">• ONE_FOR_ALL• ONE_FOR_EACH• RECORDS_STRATEGY• EMAIL_STRATEGY This inherited property is required.

Property	Datatype	Single or repeating	Description
retainer_strategy	integer	S	<p>Inherited from Retention Policy Services retainer, this indicates whether all objects are to use the same retainer or each object must have its own retainer. Valid values are:</p> <ul style="list-style-type: none"> • 0, meaning that each object must have its own retainer. • 1, meaning that all objects are to use the same retainer.

9.7 Retention Policy Services condition

Represents a template from which Retention Policy Services event objects are generated.

9.7.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_rps_condition
- Object type tag: 08

An Retention Policy Services condition object defines a condition associated with a particular phase of retention policy lifecycle. Retention Policy Services condition objects are related to phases (dmc_rps_phase_rel objects) through a relationship named dmc_rps_phase_condition_rel_type.

9.7.2 Properties

The table describes the object properties.

Table 9-6: Retention Policy Services condition type properties

Property	Datatype	Single or repeating	Description
category	string(32)	S	Specifies the category to which the condition is assigned. This is a required property.
object_name	string(255)	S	The name of the condition. For example, Project Closed or Employee Retired. This inherited property is a required property.
title	string(400)	S	User-defined description of the condition.

9.8 Retention Policy Services contact

Records information about a person defined in Retention Policy Manager.

9.8.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_rps_contact
- Object type tag: 08

Retention Policy Services contact objects contain information about persons defined in Retention Policy Manager. The objects are referenced by Retention Policy Services hold objects, Retention Policy Services authority objects, and Retention Policy Services event objects.



Note: The persons defined by Retention Policy Services contact objects are not necessarily repository users.

9.8.2 Properties

The table describes the object properties.

Table 9-7: Retention Policy Services contact type properties

Property	Datatype	Single or repeating	Description
email	string(32)	R	Email addresses for the contact.
phone	string(64)	R	Telephone numbers for the contact.
object_name	string(255)	S	Name of the contact.
title	string(400)	S	User-defined description of the contact.
user_id	ID	S	Object ID of the contact's user object in the repository if the contact is a repository user.

9.9 Retention Policy Services disposition method

Records the name of the Java class called to dispose of a retained object.

9.9.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_rps_disposition_method
- Object type tag: 08

An Retention Policy Services disposition method object identifies the Java class called to dispose of an object at the end of retention. Which class is called is dependent on the selected disposition option.

9.9.2 Properties

The table describes the object properties.

Table 9-8: Retention Policy Services disposition method type properties

Property	Datatype	Single or repeating	Description
class	string(128)	S	Path to the Java class invoked to dispose of an object.
is_container_destroyed	Boolean	S	Specifies whether the container was destroyed on disposition. T means the container is destroyed; F means that it was not.
object_name	string(255)	S	Name of the disposition method. Valid values are: <ul style="list-style-type: none">• UNKNOWN• REVIEW• TRANSFER/EXPORT• DESTROY
priority_no	integer	S	Defines the action on a retained object when two of its retainers qualify for disposition during the same qualification period. Valid values are: <ul style="list-style-type: none">• 10, undefined action• 30, review• 50, transfer or export• 70, destroy

9.10 Retention Policy Services event

Represents one instance of a dmc_rps_condition object.

9.10.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_rps_event
- Object type tag: 08

An Retention Policy Services event object is an instance of a condition defined in an Retention Policy Services condition object. It contains data specific to that instance of the condition. There is one Retention Policy Services event object generated for each Retention Policy Services condition on each Retention Policy Services phase rel object. Retention Policy Services event objects are associated with the generating Retention Policy Services condition object through a relation object named dmc_rps_condition_event_rel_type.

9.10.2 Properties

The table describes the object properties.

Table 9-9: Retention Policy Services event type properties

Property	Datatype	Single or repeating	Description
a_last_review_date	Date	S	The most recent date and time at which the condition was fulfilled.
condition_id	ID	S	Object ID of the Retention Policy Services condition object from which this event was generated. This is a required property.
fulfilled_by	ID	R	Object IDs of the Retention Policy Services contact objects representing contacts who have indicated that the event has been fulfilled.

Property	Datatype	Single or repeating	Description
fulfillment_message	string(255)	S	User-defined message to be displayed when this event is fulfilled.
object_name	string(255)	S	Name for the event. This property is required.
title	string(400)	S	User-defined description of the event.

9.11 Retention Policy Services execution rule

Defines a criterion for execution of an action.

9.11.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_rps_execution_rule
- Object type tag: 08

An Retention Policy Services execution rule object defines when a particular action is to be executed against an object controlled by a retention policy. The definition is recorded in the object name. For example, if object_name is "Phase Entry", the rule is executed only when the target object experiences phase entry.

9.11.2 Properties

The table describes the inherited object properties.

Table 9-10: Retention Policy Services execution rule type properties

Property	Datatype	Single or repeating	Description
object_name	string(255)	S	Specifies the criterion for execution of the associated action.
subject	string(192)	S	Description of the execution rule.

9.12 Retention Policy Services hold

Records information about a hold defined in Retention Policy Services.

9.12.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_rps_hold
- Object type tag: 08

An Retention Policy Services hold object represents a hold defined in Retention Policy Services. Holds prevent deletion of an object when the object is available for disposition at the end of its retention. A hold is associated with the object through a relationship named dmc_rps_hold_object_rel_type.

9.12.2 Properties

The table describes the object properties.

Table 9-11: Retention Policy Services hold type properties

Property	Datatype	Single or repeating	Description
a_last_review_date	Date	S	Date on which the status of the hold must be reviewed.
a_retention_date	Date	S	Date when the hold was applied to the object.
approvers_id	ID	R	Object IDs of the Retention Policy Services contact objects representing users who have approved the hold.
object_name	string(255)	S	Name of the hold. This is a required property.

Property	Datatype	Single or repeating	Description
requestors_id	ID	R	Object IDs of the Retention Policy Services contact objects representing the users who requested the hold. This is a required property.
title	string(400)	S	User-defined description of the hold.

9.13 Retention Policy Services notification

Specifies the recipients of notifications.

9.13.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_rps_notification
- Object type tag: 08

An Retention Policy Services notification object records the contacts to be sent notifications for a particular action and those contacts who acknowledge the notification.

9.13.2 Properties

The table describes the object properties.

Table 9-12: Retention Policy Services notification type properties

Property	Datatype	Single or repeating	Description
a_last_review_date	Date	S	Date and time at which the notification was sent.
acknowledged_by_id	ID	R	Object IDs of the Retention Policy Services contact objects representing contacts who have acknowledged the notification.

Property	Datatype	Single or repeating	Description
action_rel_id	ID	S	Object ID of the Retention Policy Services action rel object that identifies the action for a given phase.
contact_id	ID	R	Objects IDs of the Retention Policy Services contact objects representing contacts to be notified.
number_sent	integer	S	Number of times the notification has been sent.
object_name	string(255)	S	Application-defined identifier for the notification.
subject	string(192)	S	Description of the notification.

9.14 Retention Policy Services retainer

Records information about the state of an object's retention.

9.14.1 Description

- Supertype: Retainer
- Subtypes: None
- Internal name: dmc_rps_retainer
- Object type tag: 08

An Retention Policy Services retainer object mirrors an object's association with a dm_retainer object and includes additional information about the object's status in the retention lifecycle.

9.14.2 Properties

The table describes the object properties.

Table 9-13: Retention Policy Services retainer type properties

Property	Datatype	Single or repeating	Description
current_phase_id	ID	S	Object ID of the Retention Policy Services phase rel object representing the phase with which this Retention Policy Services retainer is associated.
entry_date	Date	S	Date on which the current phase was entered.
event_date	Date	S	The fulfillment date furthest in the future for the current phase's conditions.
immutability_rule	integer	S	Specifies whether objects under control of this retention policy are immutable. Valid values are: <ul style="list-style-type: none">• 0, the object is changeable.• 1, the object is immutable. This value is copied from the dm_retainer object.
object_name	string(255)	S	The object ID of the retainer. (An Retention Policy Services retainer's name is its object ID.)
parent_ancestor_id	ID	S	Reserved for future use.
phase_name	string(32)	S	Name of the phase identified in current_phase_id.

Property	Datatype	Single or repeating	Description
qualification_date	Date	S	Date on which objects in the current phase may be qualified for promotion.
r_policy_id	ID	S	Object ID of the lifecycle applied to this Retention Policy Services retainer.
retainer_root_id	ID	S	Object ID of the object attached to this Retention Policy Services retainer object. This is blank if the child strategy for the retainer is ONE_FOR_ALL. This is an inherited property.
retention_base_date	Date	S	Date from which all qualifications, promotions, and dispositions are calculated. This is a required property.
retention_date	Date	S	Earliest time at which the object attached to this retainer ceases to be retained. The date is calculated by adding sum of all phase durations to the value in retention_base_date.
retention_policy_id	ID	S	Object ID of the retainer object on which this Retention Policy Services retainer is based.
retention_rule_type	integer	S	Inherited from dm_retainer. This value must be 1, meaning retention is specified as a date.

Property	Datatype	Single or repeating	Description
ultimate_ancestor_id	ID	S	<p>Object ID of the retainer from which all other Retention Policy Services retainer objects were generated.</p> <p>This value is applicable when the child strategy is ONE_FOR_EACH.</p>
vdm_retention_rule	integer	S	<p>Defines how virtual documents and their elements are retained. Valid values are:</p> <ul style="list-style-type: none">• 0, meaning the policy may not be applied to virtual documents.• 1, meaning apply the policy only to the root document.• 2, meaning apply the policy to all virtual document components. <p>The value is copied from the retention policy when the retainer is created.</p>

9.15 Retention Policy Services retainer event rel

Associates Retention Policy Services event objects with Retention Policy Services retainer objects.

9.15.1 Description

- Supertype: State Extension
- Subtypes: None
- Internal name: dmc_rps_retainer_event_rel
- Object type tag: 37

An Retention Policy Services retainer event rel object associates Retention Policy Services event objects with a Retention Policy Services retainer object. The name of the relationship defined by Retention Policy Services retainer event rel objects is dmc_rps_retainer_event_rel_type.

9.15.2 Properties

The table describes the object properties.

Table 9-14: Retention Policy Services retainer event rel type properties

Property	Datatype	Single or repeating	Description
phase_id	ID	S	Object ID of the Retention Policy Services phase rel object to which this relationship applies.
state_no	integer	S	Specifies the phase to which the event applies.

9.16 Retention Policy Services retention policy

Defines a rule set from which Retention Policy Services retainer objects are generated.

9.16.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dmc_rps_retention_policy
- Object type tag: 08

An Retention Policy Services retention policy object defines the rules by which Retention Policy Services retainer objects are generated from the dm_retainer objects. The rules represent the influence of the retention lifecycle definition on the retention policy.

9.16.2 Properties

The table describes the object properties.

Table 9-15: Retention Policy Services retention policy type properties

Property	Datatype	Single or repeating	Description
aging_method	string(32)	S	Determines whether the promotion out of phases is based on conditions or dates. Valid values are: <ul style="list-style-type: none">• CONDITIONAL, meaning that fulfillment of conditions is used to advance through the phases.• CHRONOLOGICAL, meaning that dates are used to advance through the phases. This must be set to CONDITIONAL to define conditions for phases.
child_strategy_id	ID	S	Object ID of the Retention Policy Services child strategy object that defines the Java class invoked to apply retention.
child_strategy_class	string(128)	S	Java class that executes based on the selected child strategy. The value is a fully specified class name.
disposition_date	Date	S	Date on which objects associated with retention policy are eligible for disposition.

Property	Datatype	Single or repeating	Description
disposition_method_class	string(128)	S	Java class that executes based on the selected disposition method. The value is a fully specified class name.
disposition_method_id	ID	S	Object ID of the Retention Policy Services disposition method object that defines how to dispose of objects controlled by this retention policy.
disposition_rule	integer	S	Defines how Retention Policy Services disposes of an object. Valid values are: <ul style="list-style-type: none"> • 0, meaning DESTROY_CHILDREN • 1, meaning DESTROY_VDM • 2, meaning DESTROY_ROOT
immutability_rule	integer	S	Specifies whether objects controlled by Retention Policy Services retainers based on this policy are set to immutable. Valid values are: <ul style="list-style-type: none"> • 0, do not set the objects to immutable. • 1, set the objects immutable.
is_enabled	Boolean	S	Specifies whether the policy may be applied to objects.
object_name	string(255)	S	Name of the retention policy. This value is derived from the dm_policy on which the retention policy is based.

Property	Datatype	Single or repeating	Description
rendition_rule	integer	S	Specifies whether the retention policy applies to all content (primary and renditions) of the object or only to primary content. Valid values are: <ul style="list-style-type: none">• 0, applies to all content.• 1, applies only to primary content.
retainer.lifecycle_id	ID	S	Object ID of the lifecycle definition that is applied to all Retention Policy Services retainers generated from this retention policy object.
title	string(400)	S	User-defined description of the retention policy.
vdm_retention_rule	integer	S	Defines how virtual documents and their elements are retained. Valid values are: <ul style="list-style-type: none">• 0, meaning the retention policy may not be applied to a virtual document.• 1, meaning the policy is applied only to the root document.• 2, meaning the policy is applied to all the virtual document components.

Chapter 10

Interactive Delivery Services object reference

10.1 Interactive Delivery Services object types

The Interactive Delivery Services (earlier called as Site Caching Services) objects support Documentum Interactive Delivery Services (IDS). These object types are installed when the IDS DocApp archive file is installed in a repository.

10.2 SCS admin config

Records the common properties for all Site Caching Services (SCS) configurations in the repository and the JDBC connection information for the source repository.

10.2.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: scs_admin_config
- Object type tag: 08

10.2.2 Properties

The table describes the object properties.

Table 10-1: Scs admin config type properties

Property	Datatype	Single or repeating	Description
content_caching_enabled	Boolean	S	Indicates whether source-side caching of content is enabled. The default is F, meaning that caching is not enabled.
database_connection	string(255)	S	The JDBC URL with which to connect to the source database.

Property	Datatype	Single or repeating	Description
extra_arguments	string(255)	R	Additional arguments to be applied to all SCS configurations. The arguments are specified in a name and value pair format. For example, store_log TRUE.
object_name	string(255)	S	This property is inherited from dm_sysobject. However, because it is required for scs admin config objects, it is included in this table. For scs admin config objects, object_name is the name of the repository in which the object resides.
jdbc_driver	string(255)	S	The JDBC driver class name for the source database connection.
method_trace_level	integer	S	Specifies the method trace level for all SCS configurations in the repository. Valid values range from 0 to 10.
product_version	string(64)	S	Version level of the SCS product.

10.3 Web config

Describes the documents and properties that are to be exported to a Site Caching Services repository.

10.3.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_webc_config
- Object type tag: 08

A webc config object describes the documents and properties that are to be exported to a Site Caching Services repository.

10.3.2 Properties

The table describes the object properties.

Table 10-2: Webc config type properties

Property	Datatype	Single or repeating	Description
a_event_number	integer	S	Unique number generated internally for each publish operation.
a_increment_cnt	integer	S	Number of successful incremental refreshes since the initial publication or the last full refresh.
a_initial_publish	date	S	Date of the initial publication.
a_last_increment	date	S	Date of the last successful incremental refresh.
a_publish_status	string(128)	S	Status of the last operation.
a_refresh_date	date	S	Date of the last successful full refresh.
content_caching_enabled	Boolean	S	Indicates whether source-side caching of content is enabled. The default is F, meaning that caching is not enabled.

Property	Datatype	Single or repeating	Description
content_meta_tags	Boolean	S	If TRUE , the system inserts the document properties as metatags into the HTML files of the ContentDB. The default is FALSE.
effective_label	string(32)	S	Used to determine which documents are valid for publishing. The export operation examines the values in the a_effective_date and a_expiration_date properties of all documents whose a_effective_label property value matches the webc config effective_label property. If unspecified, effective labels are not enforced.
export_directory	string(255)	S	Local directory where exported files are placed for pickup by the transfer agent. This is a required property.
export_properties	Boolean	S	If TRUE, the system creates the PropertiesDB on the target site. If FALSE, the PropertiesDB is not created. The default is FALSE.
extra_arguments	string(255)	R	Additional arguments for this specific SCS configuration. The arguments are specified in a name and value pair format. For example, store_log TRUE.

Property	Datatype	Single or repeating	Description
is_active	Boolean	S	Indicates whether the system will publish documents described by this webc config object. TRUE (the default) directs the system to publish the documents; FALSE suspends publication.
method_trace_level	integer	S	Controls the trace level for operations. Valid values are the same as those for setServerTraceLevel method. The default is 0.
notification_user	string(255)	S	Reserved for future use.
object_name	string(255)	S	This property is inherited from dm_sysobject. However, because it is required for webc config objects, it is included in this table. For webc config objects, object_name is the logical name of the WebCache repository.
owner_name	string(255)	S	This property is inherited from dm_sysobject. For the webc config subtype, the user defined as the owner must have Superuser user privileges.

Property	Datatype	Single or repeating	Description
publish_arguments	string(255)	S	<p>Specifies arguments to pass to the publishing method. Currently, there is one valid argument: -system_validate</p> <p>This is a Boolean argument that determines, in conjunction with the target_validate_required flag in agent.ini, how authentication is conducted. -system_validate is set to T (TRUE) by default.</p> <p>Refer to the WebCache documentation for details.</p>
publish_contentless_props	Boolean	S	Indicates whether to publish documents with no content. The default is F, meaning documents without content are not published.
publish_folder_props	Boolean	S	Indicates whether to publish folder properties. The default is F, meaning that folder properties are not published.
send_notification	Boolean	S	Reserved for future use.
source_attr_isrep	Boolean	R	Indicates whether the property named in the corresponding index position in sourceAttrs is a repeating property. TRUE means the property is repeating. FALSE means the property is single-valued.

Property	Datatype	Single or repeating	Description
source_attr_length	integer	R	Length of the property named in the corresponding index position in sourceAttrs.
source_attr_type	integer	R	Specifies the datatype of the property named in the corresponding index position in sourceAttrs. Valid values are: <ul style="list-style-type: none"> • 0, for Boolean • 1, for Integer • 2, for String • 3, for ID • 4, for Date • 5, for Double
sourceAttrs	string(80)	R	Specifies additional properties to export to the PropertyDB. The specification syntax is: <i>type_name.property_name</i> If this property has no values, then only the default properties are exported.
source_folder_id	ID	S	Object ID of the local folder that is the root of the repository structure that matches the WebCache repository. This is a required property.

Property	Datatype	Single or repeating	Description
source_formats	string(64)	R	<p>Document formats to publish.</p> <p>If formats are specified in this property, only documents in the specified formats are published. If a document has multiple formats (a primary format and renditions), only those that are included in the list are published.</p> <p>If this property is unspecified, all existing renditions, including the one in the primary format, are published by default.</p>
source_version	string(32)	S	Version of the WebCache software running on the source. (Informational only)
target_id	ID	R	<p>Object ID of the webc target object that references this webc config object. This is a required property.</p> <p>This is a repeating property to allow for future enhancements.</p>
transfer_method	string(32)	S	Reserved for future use.

Property	Datatype	Single or repeating	Description
version_labels	string(32)	R	<p>Specifies the repository version you want to publish. Only documents carrying the specified version are published. Specify only one version label.</p> <p>If you set this to ANY VERSION, the system publishes whichever version is found in the source folder (the source folder is identified in source_folder_id). It is assumed that only one version of any particular document will be in the source folder at a given time.</p> <p>The default is CURRENT.</p> <p> Note: This is a repeating property to allow for future enhancements.</p>

10.4 Webc target

Describes a Site Caching Services repository on a Site Caching Services target host.

10.4.1 Description

- Supertype: SysObject
- Subtypes: None
- Internal name: dm_webc_target
- Object type tag: 08

A webc target object describes a Site Caching Services repository on a Site Caching Services target host.

10.4.2 Properties

The table describes the object properties.

Table 10-3: Webc target type properties

Property	Datatype	Single or repeating	Description
app_server	string(32)	S	Name of the Web site delivery engine. (Informational only).
app_server_version	string(32)	S	Version level of the Web site delivery engine. (Informational only).
extra_arguments	string(255)	R	Reserved for future use.
global_locales	string(5)	R	Specifies the locales for global publishing. This is only set if global_publishing is set to T.
global_publishing	Boolean	S	Indicates whether global publishing is enabled. The default is F, meaning global publishing is not enabled.
object_name	string(255)	S	This property is inherited from dm_sysobject. However, because it is required for webc target objects, it is included in this table. For webc target objects, object_name is the logical name of the WebCache repository.

Property	Datatype	Single or repeating	Description
online_sync	Boolean	S	Indicates whether the data in the WebCache repository is accessible to users during the synchronization operation. If set to TRUE, the synchronization agent minimizes the possibility that users will be interrupted by repository updates. The default is FALSE.
online_sync_dir	string(255)	S	Remote directory used for the backup copy of the WebCache repository during online updates. This is required if online_sync is TRUE.
owner_name	string(255)	S	This property is inherited from dm_sysobject. For the webc target subtype, the user defined as the owner must have Superuser user privileges.
post_sync_script	string(48)	S	Name of a script to be executed after synchronization. The script must be stored in the WebCache bin directory on the target machine. Refer to the WebCache documentation for details.

Property	Datatype	Single or repeating	Description
pre_sync_script	string(48)	S	Name of a script to be executed prior to synchronization. The script must be stored in the WebCache bin directory on the target machine. Refer to the WebCache documentation for details.
probdb_flags	string(64)	R	Reserved for future use.
propdb_tablename	string(64)	S	Name to use when creating the PropertyDB tables. Three tables are created using the name in this property: <i>propdb_tablename_s</i> <i>propdb_tablename_r</i> <i>probdb_tablename_m</i> You must specify a name if the export_properties property in the webc config object is set to TRUE.
propdb_dbversion	string(32)	S	Version level of the target RDBMS. (Informational only)

Property	Datatype	Single or repeating	Description
secure_connection	string(16)	S	<p>Specifies which agent port to use for the transfer. Valid values are:</p> <ul style="list-style-type: none"> • ssl, meaning use the target_ssl_port • raw, meaning use the target_raw_port • both, meaning use the target_ssl_port if available; if not, use the target_raw_port <p>Documentum Administrator will not allow you to specify the both option; it must be set through the API. If you publish using the both option, you will receive warning messages in the log file.</p>
sync_arguments	string(128)	R	Reserved for future use.
target_host	string(128)	S	Host name or IP address of the WebCache host machine.
target_is_active	Boolean	S	Reserved for future use.
target_raw_port	integer	S	<p>Port number at which the target Agent process resides. This port is used for unencrypted transmissions.</p> <p>Either this property or target_ssl_port must be set.</p>

Property	Datatype	Single or repeating	Description
target_root_directory	string(255)	S	The physical directory on the Web server where the files will be placed. This is a required property.
target_ssl_port	integer	S	Port number at which the target Agent process resides. This port is used for encrypted transmissions. Either this property or target_raw_port must be set.
target_version	string(32)	S	Version level of the WebCache software running on the host server. (Informational only)
target_virtual_dir	string(255)	S	The URL for the WebCache installation. This is a required property if you are using WebPublisher.
transfer_arguments	string(128)	R	Reserved for future use.
transfer_directory	string(255)	S	Remote directory where exported files are placed for pickup by the Synchronization agent. This property is required if the transfer_method property in the webconfig object is set.
transfer_domain	string(16)	S	For Windows NT, the domain of the user identified in transfer_user. For other platforms, this is unused.
transfer_protocol	string(16)	S	Reserved for future use.

Property	Datatype	Single or repeating	Description
transfer_user	string(255)	S	User name to use to access the remote host in the transfer operation. The default is the local repository owner.
transfer_user_password	string(255)	S	Password for the user identified in transfer_user.

Chapter 11

Forms Builder object reference

11.1 Forms Builder object types

Forms Builder object types support Documentum Forms Builder. The object types are installed with the Documentum Forms Builder application.

11.2 XFM adaptor config

Records information about a forms adaptor.

11.2.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dmc_xfm_adaptor_config
- Object type tag: 09

An xfm adaptor config object stores information about a forms adaptors. An adaptor is executable Java code invoked by the Forms engine, which allows a form to interact with external data sources at runtime. Instances of the type are created internally when Documentum Forms Builder is used to create a form template.

11.2.2 Properties

The describes the object properties.

Table 11-1: Xfm adaptor config type properties

Property	Datatype	Single or repeating	Description
adaptor_type	string(16)	S	<p>Specifies the type of adaptor. Valid values are:</p> <ul style="list-style-type: none">• datasource• item-intializer• item-validator• doc-validator• doc-processor

Property	Datatype	Single or repeating	Description
impl_class	string(255)	S	The implementation class, with a fully qualified namespace.
impl_type	string(16)	S	Class name of the SBO.
input_name	string(64)	R	Name of an input.
input_required	Boolean	R	Specifies whether an input is required. T means an input is required. The value at a particular index position is applied to the input specified at the corresponding position in input_name.
input_type	string(16)	R	Type of an input. The type specified at a particular index position defines the input specified at the corresponding index position in input_name.
is_public	Boolean	S	Specifies whether this adaptor is public. T means the adaptor is public. F means the adaptor is not public.
output_type	string(64)	S	If the output is a primitive type, it can be string, Boolean, integer, date, and so on. If the output is a complex type, the value is the name of the complex type.
param_name	string(64)	R	Names of the initialization parameters.

Property	Datatype	Single or repeating	Description
param_value	string(255)	R	Values of the initialization parameters. The value at a particular index position is the value of the initialization parameter identified at the corresponding index position in param_name.

11.3 XFM form

Contains information about a form template.

11.3.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dm_xfm_form
- Object type tag: 09

An xfm form object stores information about a form template. Instances of the type are created internally when Documentum Forms Builder is used to create a form template.

11.3.2 Properties

The table describes the object properties.

Table 11-2: Xfm form type properties

Property	Datatype	Single or repeating	Description
definition_state	integer	S	Indicates the status of the form. Valid values are: <ul style="list-style-type: none"> • 0, meaning draft • 1, meaning validated • 2, meaning installed • 3, meaning obsolete

Property	Datatype	Single or repeating	Description
Description	string(300)	S	User-defined description of the form.
display_mode	integer	S	Indicates the display mode of the form. Valid values are: <ul style="list-style-type: none">• 0, meaning dialog• 1, meaning publish
relationship_to_base	integer	S	Specifies the relationship between the current form and the base form. Valid values are: <ul style="list-style-type: none">• 0, meaning the form is not related to the base form• 1, meaning the form is related to the base form

11.4 XFM instance

Represents an instance of a form.

11.4.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dm_xfm_instance
- Object type tag: 09

A xfm instance object contains information about one form. Instances of the type are created internally when Documentum Forms Builder is used to create a form template.

11.4.2 Properties

The table describes the object properties.

Table 11-3: Xfm instance type properties

Property	Datatype	Single or repeating	Description
is_default	Boolean	S	<p>Indicates the form is a form default instance.</p> <p>T (TRUE) means the form is a default instance. F (FALSE) means the form is not a default instance.</p>

11.5 XFM schema

Stores information about an XML schema for a form.

11.5.1 Description

- Supertype: dm_document
- Subtypes: None
- Internal name: dm_xfm_schema
- Object type tag: 09

An xfm schema object represents an XML schema used with a form. Instances of the type are created internally when Documentum Forms Builder is used to create a form.

11.5.2 Properties

The table describes the object properties.

Table 11-4: Xfm schema type properties

Property	Datatype	Single or repeating	Description
data_model	string(128)	S	Specifies the root element of the schema.

Property	Datatype	Single or repeating	Description
storage_option	integer	S	<p>Specifies the storage mode for the schema. Valid values are:</p> <ul style="list-style-type: none">• 0, meaning store as content.• 1, meaning store as content and use an XML Application to populate object properties with the content.
storage_type	string(32)	S	Name of the object type of the object with which the content of the form instance is associated.

Chapter 12

Editorial Publishing Service type reference

12.1 Editorial Publishing Services types

The editorial publishing services objects are added when the Documentum Content Services for Woodwing product is installed. The Woodwing Smart Connection product interacts with a Documentum CM Server repository.

12.2 EPS config cache

Records configuration caching information

12.2.1 Description

- Supertype: None
- Subtypes: None
- Internal name: dmc_eps_config_cache
- Object type tag: 00

An eps config cache object records configuration caching information. This information is used internally to manage cached information about specific object type configurations.



Caution

Do not modify or delete this type or instances of this type.

12.2.2 Properties

The table describes the object properties.

Table 12-1: Eps config cache type properties

Property	Datatype	Single or repeating	Description
eps_object_type	string(32)	S	Name of the EPS object type.
eps_time_stamp	Time	S	Time stamp of the object type's last modification.

12.3 EPS elem type mapping

Records the object types to use with a particular element type

12.3.1 Description

- Supertype: EPS Mapping
- Subtypes: None
- Internal name: dmc_eps_elem_type_mapping
- Object type tag: 0b

An object type eps elem type mapping records the object types that may be used with a particular element type. Objects of this type are created and managed internally.

12.3.2 Properties

The table describes the object properties.

Table 12-2: Eps elem type mapping type properties

Property	Datatype	Single or repeating	Description
eps_elem_type	string(32)	S	Element type to which the object type is mapped.
eps_object_type_name	string(32)	S	Object type mapped to the element type identified in eps_elem_type.

12.4 EPS element

Serves as the base object type for all element subtypes.

12.4.1 Description

- Supertype: Document
- Subtypes: None
- Internal name: dmc_eps_element
- Object type tag: 09

This object type serves as the base object type for all element object types. All element object types are subtypes of eps element and inherit its properties. Those properties record basic information about an element.

Users may create subtypes of the object type through the Content Services for Woodwing object type editor in the Publication Manager.

12.4.2 Properties

The table describes the object properties.

Table 12-3: Eps element type properties

Property	Datatype	Single or repeating	Description
eps_creation_action	string(32)	S	Creation action type used to create the element.
eps_element_type	string(32)	S	Type of element.
eps_issue_id	ID	S	Object ID of the issue that contains the element.
eps_publication_id	ID	S	Object ID of the publication that contains the element.
eps_route_to_owners	string(2000)	S	Names of users who have the element in their inbox.
eps_section_id	ID	S	Object ID of the section that contains the element.
eps_state_name	string(255)	S	Name of the workflow state the element is in.
eps_workflow_id	ID	S	Object ID of the workflow.
eps_workflow_template_id	ID	S	Object ID of the workflow template that was the source of the workflow in eps_workflow_id.

12.5 EPS issue

Serves as the base object type for issue object types

12.5.1 Description

- Supertype: Folder
- Subtypes: None
- Internal name: dmc_eps_issue
- Object type tag: 0b

This object type serves as the base object type for all issue object types. All issue object types are subtypes of eps issue and inherit its properties. Those properties record basic information about an issue.

12.5.2 Properties

The table describes the object properties.

Table 12-4: Eps issue type properties

Property	Datatype	Single or repeating	Description
eps_archived	Boolean	S	Specifies whether the object is archived.
eps_issue_deadline	Date	S	Date on which issue is to be released.
eps_publication_id	ID	S	Object ID of the publication to which this issue belongs.

12.6 EPS mapping

Serves as the base object type for all mapping tables.

12.6.1 Description

- Supertype: None
- Subtypes: EPS Metadata Mapping, EPS State Perm Mapping, EPS Performer Mapping, EPS Template Mapping, EPS Schedule Mapping
- Internal name: dmc_eps_mapping
- Object type tag: 00

This object type records mapping information common to all publications. It serves as the base type for all other mapping related object types.

**Caution**

Do not modify or delete this object type or instances of the type.

12.6.2 Properties

The table describes the object properties.

Table 12-5: Eps mapping type properties

Property	Datatype	Single or repeating	Description
eps_issue_id	ID	S	Issue to which the mapping applies.
eps_publication_id	ID	S	Publication to which the mapping applies.
eps_section_name	string(255)	S	Name of the section to which this mapping applies.

12.7 EPS metadata mapping

Records the properties visible on each Woodwing dialog box.

12.7.1 Description

- Supertype: EPS Mapping
- Subtypes: None
- Internal name: dmc_eps_metadata_mapping
- Object type tag: 0b

An eps metadata mapping object records the properties of an element that are exposed on a Woodwing dialog box. Objects of this type are created and managed internally.

12.7.2 Properties

The table describes the object properties.

Table 12-6: Eps metadata mapping type properties

Property	Datatype	Single or repeating	Description
eps_attribute_names	string(255)	R	Names of the visible properties.
eps_element_type	string(32)	S	Element type to which this object applies.
eps_woodwing_dialog	string(32)	S	Name of the Woodwing dialog box on which these properties are visible.

12.8 EPS page detail

Records information about a page in an element in a publication.

12.8.1 Description

- Supertype: None
- Subtypes: None
- Internal name: dmc_eps_page_detail
- Object type tag: 00

An eps page detail object records information about a page within an element.



Caution

Do not modify or delete this type or instances of this type.

12.8.2 Properties

The table describes the object properties.

Table 12-7: Eps page detail type properties

Property	Datatype	Single or repeating	Description
eps_element_id	ID	S	Object ID of the element for which this object records page details.
eps_page_height	string(32)	S	Page height.
eps_page_number	string(32)	S	Page number.
eps_page_order	string(32)	S	Page order.
eps_page_width	string(32)	S	Page width.

12.9 EPS performer mapping

Records performer information.

12.9.1 Description

- Supertype: EPS Mapping
- Subtypes: None
- Internal name: dmc_eps_performer_mapping
- Object type tag: 0b

An eps performer mapping object records information about a particular performer. Objects of this type are created and managed internally.

12.9.2 Properties

The table describes the object properties.

Table 12-8: Eps performer mapping type properties

Property	Datatype	Single or repeating	Description
eps_performer	string(255)	S	Identifies the performer to whom this mapping applies.
eps_performer_can_revert	Boolean	S	Specifies whether the performer can revert object states.

Property	Datatype	Single or repeating	Description
eps_performer_group_name	string(255)	S	Name of the group that records performer users.

12.10 EPS placement

Records information about placed articles and the metadata associated with the link.

12.10.1 Description

- Supertype: None
- Subtypes: None
- Internal name: dmc_eps_placement
- Object type tag: 00

An eps placement object records information about placed articles and the metadata associated with the link.



Caution

Do not modify or delete this type or instances of this type.

12.10.2 Properties

The table describes the object properties.

Table 12-9: Eps placement type properties

Property	Datatype	Single or repeating	Description
eps_child_id	ID	S	Object ID of the child in the placement.
eps_parent_id	ID	S	Object ID of the parent in the placement.
eps_placement_props	string(255)	R	Properties of the placement.

12.11 EPS pubinfo cache

Records the last modification date of a publication

12.11.1 Description

- Supertype: None
- Subtypes: None
- Internal name: dmc_eps_pubinfo_cache
- Object type tag: 00

An eps pubinfo cache object records the last modification time of a publication. Objects of this type are used internally to manage publications.



Caution

Do not modify or delete this type or instances of this type.

12.11.2 Properties

The table describes the object properties.

Table 12-10: Eps pubinfo cache type properties

Property	Datatype	Single or repeating	Description
eps_publication_id	ID	S	Object ID of the publication.
eps_last_mod_time	integer	S	The time of the publication's last modification.

12.12 EPS publication

Serves as the base object type for all publication object types.

12.12.1 Description

- Supertype: Folder
- Subtypes: None
- Internal name: dmc_eps_publication
- Object type tag: 0b

This object type serves as the base object type all publication object types. All publication object types are subtypes of eps publication and inherit its properties. Those properties record basic information about a publication.

12.12.2 Properties

The table describes the object properties.

Table 12-11: Eps publication type properties

Property	Datatype	Single or repeating	Description
eps_archived	Boolean	S	Specifies whether the publication is archived.
eps_section_names	string(255)	R	Names of the sections in the publication.

12.13 EPS relation

Associates articles, layouts, and images.

12.13.1 Description

- Supertype: Relation
- Subtypes: None
- Internal name: dmc_eps_relation
- Object type tag: 37

An eps relation object is used to establish the links between an article and any images it contains and between the article and its layout. The objects are created automatically as needed.

12.13.2 Properties

The table describes the object properties.

Table 12-12: Eps relation type properties

Property	Datatype	Single or repeating	Description
eps_child_version	string(32)	S	Version of the child object used in the relationship.
eps_parent_version	string(32)	S	Version of the parent object used in the relationship.
eps_placement_type	string(255)	S	Specifies whether this represents a native relationship or a placeholder relationship.

12.14 EPS section

Serves as the base object type for all section object types.

12.14.1 Description

- Supertype: Folder
- Subtypes: None
- Internal name: dmc_eps_section
- Object type tag: 0b

This object type serves as the base object type for all section object types. All section object types are subtypes of eps section and inherit its properties. Those properties record basic information about a sections.

12.14.2 Properties

The table describes the object properties.

Table 12-13: Eps section type properties

Property	Datatype	Single or repeating	Description
eps_issue_id	ID	S	Object ID of the issue that contains this section.

Property	Datatype	Single or repeating	Description
eps_publication_id	ID	S	Object ID of the publication that contains this section.

12.15 EPS state

Records information about states.

12.15.1 Description

- Supertype: None
- Subtypes: None
- Internal name: dmc_eps_state
- Object type tag: 00

An eps state object records information about states in an editorial workflow process.



Caution

Do not modify or delete this type or instances of this type.

12.15.2 Properties

The table describes the object properties.

Table 12-14: Eps state type properties

Property	Datatype	Single or repeating	Description
eps_produce_pdf_flag	Boolean	S	Default value for the produce PDF flag for this element.
eps_required_permission	integer	S	Permission level required to perform the task associated with this state.
eps_state_color	string(32)	S	State color.
eps_state_duration	integer	S	Expected duration of the state, in days.
eps_state_name	string(255)	S	Name of the state.

Property	Datatype	Single or repeating	Description
eps_state_permissions	string(300)	R	List of performer types and their permissions.

12.16 EPS state mapping

Records state configuration information for publications.

12.16.1 Description

- Supertype: EPS Mapping
- Subtypes: None
- Internal name: dmc_eps_state_mapping
- Object type tag: 0b

An eps state mapping object records configuration information about a state for a particular element in a publication. Objects of this type are created and managed internally.

12.16.2 Properties

The table describes the object properties.

Table 12-15: Eps state mapping type properties

Property	Datatype	Single or repeating	Description
eps_creation_action	string(32)	S	The creation action type to which this state information applies.
eps_element_type	string(32)	S	The element type to which this state information applies.
eps_produce_pdf	Boolean	S	Specifies whether to generate a PDF preview of the elements in this state.
eps_state_color	string(32)	S	Color to associate with elements in this state.

Property	Datatype	Single or repeating	Description
eps_state_duration	integer	S	Expected length of time, from start to finish, for the completion of the state.
eps_state_name	string(255)	S	Name of the state.

12.17 EPS state perm mapping

Records the permissions held by performers for each state.

12.17.1 Description

- Supertype: EPS Mapping
- Subtypes: None
- Internal name: dmc_eps_state_perm_mapping
- Object type tag: 0b

An eps state perm mapping object records the permission held by a performer for a particular state in an editorial workflow process. Objects of this type are created and managed internally.

12.17.2 Properties

The table describes the object properties.

Table 12-16: Eps state perm mapping type properties

Property	Datatype	Single or repeating	Description
eps_creation_action	string(32)	S	The creation action type to which the permission applies.
eps_element_type	string(32)	S	The element type to which the permission applies.
eps_performer	string(255)	S	Performer who holds this permission.
eps_performer_permission	integer	S	Permission level assigned to the performer identified in epss_performer.

Property	Datatype	Single or repeating	Description
eps_state_name	string(255)	S	Name of the state to which this permission applies.

12.18 EPS state transition

Records all state transitions.

12.18.1 Description

- Supertype: None
- Subtypes: None
- Internal name: dmc_eps_state_transition
- Object type tag: 00

An eps state transition object records a state transition. These objects are used internally, for reporting on issue and element status.



Caution

Do not modify or delete this type or instances of this type.

12.18.2 Properties

The table describes the object properties.

Table 12-17: Eps state transition type properties

Property	Datatype	Single or repeating	Description
eps_element_id	ID	S	Object ID of the element participating in the state transition.
eps_end_state	string(32)	S	Name of the end state in the transition.
eps_start_state	string(32)	S	Name of the start state in the transition.
eps_time_stamp	string(32)	S	Time stamp of the transition.

12.19 EPS template mapping

Records the templates to use for element types.

12.19.1 Description

- Supertype: EPS Mapping
- Subtypes: None
- Internal name: dmc_eps_template_mapping
- Object type tag: 0b

An eps template mapping object records the template to be used for a particular element type. Objects of this type are created and managed internally.

12.19.2 Properties

The table describes the object properties.

Table 12-18: Eps template mapping type properties

Property	Datatype	Single or repeating	Description
eps_element_type	string(32)	S	Name of the element type.
eps_template_id	ID	S	Object ID of the template object representing the template to be used with the element identified in eps_element_type.

12.20 EPS transition override

Records scheduling changes to the master schedule configured for an issue

12.20.1 Description

- Supertype: None
- Subtypes: None
- Internal name: dmc_eps_transition_override
- Object type tag: 00

An eps transition override object records a schedule change for an element within an issue.

**Caution**

Do not modify or delete this type or instances of this type.

12.20.2 Properties

The table describes the object properties.

Table 12-19: Eps transition override type properties

Property	Datatype	Single or repeating	Description
eps_element_id	ID	S	Object ID of the element participating in the state transition.
eps_state	string(255)	S	Name of the state in the transition.
eps_start_duration	integer	S	Duration of the state.
eps_workflow_template_id	ID	S	Object ID of the workflow template.

12.21 EPS WF template mapping

Specifies the workflow template to be used with an element and creation action type.

12.21.1 Description

- Supertype: EPS Mapping
- Subtypes: None
- Internal name: dmc_eps_wf_template_mapping
- Object type tag: 0b

An eps wf template mapping object specifies which workflow template is to be used with a particular combination of element type and creation action type. Objects of this type are created and managed internally.

12.21.2 Properties

The table, describes the object properties.

Table 12-20: Eps wf template mapping type properties

Property	Datatype	Single or repeating	Description
eps_creation_action	string(32)	S	The creation action type with which the workflow template is associated.
eps_element_type	string(32)	S	Name of the element type.
eps_workflow_template_id	ID	S	Object ID of the workflow template object representing the workflow template to be used with the element identified in eps_element_type.

12.22 EPS workflow data

Records information to be passed to the Java workflow method.

12.22.1 Description

- Supertype: None
- Subtypes: None
- Internal name: dmc_eps_workflow_data
- Object type tag: 00

An eps workflow data object records information that is passed to a Java workflow method.



Caution

Do not modify or delete this type or instances of this type.

12.22.2 Properties

The table describes the object properties.

Table 12-21: Eps workflow data type properties

Property	Datatype	Single or repeating	Description
eps_element_id	ID	S	Object ID of the element to which this information applies.
eps_performer_name	string(255)	S	Name of the performer to whom the task is assigned or delegated.
eps_state_name	string(255)	S	Name of the state to set the object to.
eps_workflow_id	ID	S	Object ID of the workflow to which this information is passed.

Appendix A. Language and country codes

A.1 Language and country codes

This appendix lists the recommended language codes and some of the recommended country codes for use in the language_code property of SysObjects. The language codes are taken from ISO 639. The country codes are taken from the larger set found in ISO 3116.

Table A-1: Recommended language codes

Language	Code	Language	Code
Albanian	sq	Arabic	ar
Armenian	hy	Azerbaijani	az
Basque	eu	Bulgarian	bg
Byelorussian	be	Cambodian	km
Catalan	ca	Chinese	zh
Croatian	hr	Czech	cs
Danish	da	Dutch	nl
English	en	Estonian	et
Finnish	fi	French	fr
Georgian	ka	German	de
Greek	el	Hebrew	he
Hungarian	hu	Icelandic	is
Indonesian	id	Italian	it
Japanese	ja	Kazakh	kk
Korean	ko	Latvian, Lettish	lv
Lithuanian	lt	Macedonian	mk
Malay	ms	Moldavian	mo
Norwegian	no	Persian	fa
Polish	pl	Portuguese	pt
Romanian	ro	Russian	ru
Serbian	sr	Serbo-Croatian	sh
Slovak	sk	Slovenian	sl
Spanish	es	Swedish	sw
Thai	th	Turkish	tr
Ukrainian	uk	Uzbek	uz

Language	Code	Language	Code
Vietnamese	vi		

Table A-2: Recommended country codes

Country	Code	Country	Code
Argentina	AR	Austria	AT
Australia	AU	Azerbaijan	AZ
Belarus	BY	Belgium	BE
Belize	BZ	Bolivia	BO
Brazil	BR	Bulgaria	BG
Cambodia	KH	Cameroon	CM
Canada	DA	Chile	CL
China	CN	Colombia	CO
Costa Rica	CR	Croatia	HR
Czech Republic	CZ	Denmark	DK
Ecuador	EC	Egypt	EG
Estonia	EE	Finland	FI
France	FR	Gabon	GA
Gambia	GM	Georgia	GE
Germany	DE	Great Britain (UK)	GB
Greece	GR	Greenland	GL
Guatemala	GT	Haiti	HT
Honduras	HN	Hong Kong	HK
Hungary	HU	Iceland	IS
India	IN	Indonesia	ID
Iran	IR	Iraq	IQ
Ireland	IE	Israel	IL
Italy	IT	Japan	JP
Jordan	JO	Kazachstan	KZ
Kenya	KE	Korea (North)	KP
Korea (South)	KR	Kyrgyz Republic	KG
Laos	LA	Latvia	LV
Lebanon	LB	Libya	LY
Liechstenstein	LI	Lithuania	LT

Country	Code	Country	Code
Luxemborg	LU	Mexico	MX
Mozambique	MZ	Morocco	MA
Netherlands	NL	New Zealand	NZ
Nicaragua	NI	Norway	NO
Pakistan	PA	Panama	PK

Appendix B. RDBMS tables

B.1 RDBMS tables for OpenText Documentum CM types

All Documentum CM Server objects and their property values are stored in a repository. A repository is part of a larger relational database (RDBMS).

Within the relational database, each object type is represented by two tables and two views:

- A table and a view for single-valued object properties
- A table and a view for repeating object properties.



Note: The example tables assume that the underlying RDBMS database is an Oracle database.

The name of the table describing an object's single-valued properties has the format: *typename_s*. For example, the table that contains the single-valued properties for the document type is the *dm_document_s* table.

The names of the tables describing the repeating properties have the following format: *typename_r*. For example, the table that contains the repeating properties of the document type is the *dm_document_r* table.

The views join the a type's tables with the tables of its supertypes. For example, the view of the single-valued properties for folders joins the *dm_folder_s* and *dm_sysobject_s* tables. For cabinets, the view joins the *dm_cabinet_s*, *dm_folder_s*, and *dm_sysobject_s* tables. The names of the views that are built on the single-valued property tables have the following format: *typename_sp*. For example, the single-valued property view associated for *dm_document* type is named *dm_document_sp*.

➤ Example B-1: User type table examples

The examples describe *_s* and *_r* tables, with the *dm_user* type as the demonstrative type. The views do not differ for this type because it has no supertypes.

The following statement retrieves the columns of the *dm_user_s* table:

```
sql>describe dm_user_s
```



Table B-1: Columns of the dm_user_s table

Column name	Datatype	Nullable (Yes/No)
R_OBJECT_ID	CHAR(16)	N
GROUP_NAME	CHAR(255)	N
GROUP_ADDRESS	CHAR(80)	N
I_VSTAMP	NUMBER(10)	N
I_ATTR_EXTRA2	CHAR(64)	Y
I_ATTR_INT1	NUMBER(10)	Y
I_ATTR_INT2	NUMBER(10)	Y
I_VSTAMP	NUMBER(10)	Y

The following statement retrieves a description of the dm_user_r table:

```
sql>describe dm_user_r;
```

Table B-2: Columns of the dm_user_r table

Column name	Datatype	Nullable (Yes/No)
R_OBJECT_ID	NUMBER(10)	N
I_POSITION	NUMBER(6)	N
USER_NAMES	CHAR(255)	Y
GROUPS_NAMES	CHAR(255)	Y
I_ALL_USERS_NAMES	CHAR(16)	Y

Appendix C. Obsolete, deprecated, or unused objects

C.1 Obsolete, deprecated, or unused object types

This appendix lists object types that are obsolete, deprecated, or unused. An obsolete type was used in prior releases to support some functionality. An unused object type was added to the hierarchy, but never used by Documentum CM Server or any OpenText Documentum CM client products.

The following object types are obsolete or unused:

- dm_java
- dm_staged
- dm_router

The following object types are supported but deprecated because they have been superceded by another object type or are used by deprecated features or products:

- dm_format_preferences

This object type is used by Desktop Client to record information about which rendition and application to use for editing and viewing a document in a particular format.

- dm_menu_system

This object type is used by Desktop Client to record the Explorer Integration menu configuration.

- dm_email_message

The reference information for this type is still documented.

