



Core

Administrator guide

Référence : 52 CORE-ADM CORE.V1

Ever Team

336 rue Saint-Honoré - 75001 PARIS - France

17 quai Joseph Gillet - 69316 LYON Cedex 04 - France

33 (0)1 72 71 33 00 • www.everteam.com

Terms and conditions

Documentation

© Ever Team, 2016



Attribution No Derivatives
cc by-nd

This documentation is made available under a Creative Commons license.

This license permits you to copy, distribute, display and make text copies only of your work, but not of any derivative works.

[To view a copy of this license](#) | [View license](#)

everteam software

The right to use the everteam™ software package, the subject of this documentation, is governed by the general and specific terms of the contract for use of Ever Team software.

everteam™ is a registered trademark of Ever Team; all other trademarks mentioned herein are registered by their owners, and are their property.

All rights reserved for all countries. Original software developed by Ever Team.

Table of contents

1	Introduction	5
2	Parameters	6
2.1	Menus	6
2.1.1	Core	6
2.1.2	Services	6
2.1.3	User Interface	6
2.1.4	Individual document scanning	6
2.1.5	Outgoing email	7
2.1.6	Parameter table	8
2.1.7	Parameters in use	8
2.2	Review of how parameters are saved	8
2.3	Core parameters	9
2.3.1	Processing at connection time	9
2.3.2	Core parameters list	10
2.3.3		19
3	Management	19
3.1	Refresh	19
3.2	Dictionary	20
3.2.1	Basics	20
3.2.2	Loading and priority	21
3.3	Forms	21
3.3.1	Basics	21
3.3.2	Loading	22
3.4	Locks	22
3.5	Do links	23
3.5.1	Basic concepts	23
3.5.2	Execution	24
4	Technical administration	25
4.1	everteam variables	25
4.2	Token creation	26
4.3	Thread pools	26
4.3.1	Basic concepts	27
4.3.2	ThreadPools	27
4.3.3	Job families	27
4.3.4	Execution groups	28
4.3.5	Mappings	28
4.3.6	Monitor	29
5	Monitoring	30
5.1	Monitor	30
5.2	Scheduler	31

5.3 Update counters	31
5.3.1 Basic concepts	31
5.3.2 Execution	31
5.4 Audit	31
5.4.1 Activity statistics	32
5.4.2 Data import	32
5.4.3 Temporary files	33
5.4.4 Documents	33
6 Appendices	35
6.1 LockManager class	35

1 Introduction

The infrastructure is the foundation of any everteam application. An application is implemented and administered via the standard everteam web interface web.

This guide explains how to manage the everteam infrastructure.

The menus used are the following:



[Parameters]: access application parameters (Core, Services, User interface, Individual document scanning, Outgoing email, Parameter table).



[Management]: Refresh parameters, Manage the dictionary, forms and record locks, Rebuild links between tables.



[Technical administration]: Execute SQL (to run advanced queries), List variables, Create a token, Manage thread pools).



[Monitoring]: Monitor and analyze the server, List schedulers with their status, Update everteam counters, Run an application audit.

Reference

The deployment and management of an application and its databases are explained in the CORE Admin - Application and data guide. The menus discussed there are: Data source and Functional administration.

The Security menu is described in the CORE Security guide.

Reference

The CORE Standard user interface guide explains how to set up and manage the user interface. This guide covers configuring the "ihm" service and associated features.

2 Parameters

The [everteam.core > Parameters](#) menu offers several entry points to parameter setup, depending on the context.

2.1 Menus

2.1.1 Core

This menu is for declaring all the everteam Core parameters for the application.

2.1.2 Services

This menu centralizes access to the setup of all the everteam services used by the application.

Reminder: each service has its own parameters. These parameters are generally declared in the menu of each service. In each service, the [Parameters](#) menu lists the parameters to set so that the service works as required in your environment. The list of parameters is identical there to that found in the sub-menu described below.

Each service also comes with a [param.xml](#) file which contains its default parameters, and in particular any new parameters for new features in the service.

For each service, the parameters are defined in: `./apps/Service_name/param.xml`

For example, for the et.ais service: `./apps/ais/conf/param.xml`.

These parameters may be used in forms (or *templates*) as static variables. The syntax to call them is:

```
${sd.parameter_name}
```

Example: extract from a form. This is the definition of a variable (vUrl) which uses the standard variable "\${sd.dPortalDBName}" to get the name of the system database.

```
var vUrl = "GetRecords? Template=topicmap/tm & DBName=${sd.dPortalDBName}"
```

2.1.3 User Interface

This menu is for managing the interface parameters, for all users, and on all tables.

2.1.4 Individual document scanning

This menu is for managing digitization parameters (see [Advanced parameters > Individual document scanning](#) menu).

Reference

Please refer to the *SCU Individual Document Scanning Guide*.

2.1.5 Outgoing email

This menu is for configuring sending emails from everteam (SMTP, default values, etc.).

Outgoing email settings

SMTP

Host

Port

User

Password

Default values

From

To

Subject

Send email ☒

Store emails ☐

Target

Service

Service ☐ ☐

Outgoing emails menu

Default values are stored in the [mailer](#) core parameter.

In the [Service](#) section, a click on [\[Specify a service\]](#) allows you to define specific values for the everteam.archive solution and/or the services available in the dropdown list.

Service

Service ☐ ☐

Outgoing emails menu /Specify a service

Access to these features depends on some core security rights: [CORE_CONFIGMAIL.SHOWCONFIG](#) (Show outgoing email settings) and [CORE_CONFIGMAIL.SAVECONFIG](#) (Save outgoing email settings). These can be managed in the [everteam.core > Security > Service security > everteam.core](#) menu.

2.1.6 Parameter table

This menu provides the full list of parameters stored in the [CSPARAMS](#) table, necessary for everteam and its various applications to function. There you find some general parameters which do not depend on any service. A user can modify them if he has been given the rights.

How these parameters are used depends on their type:

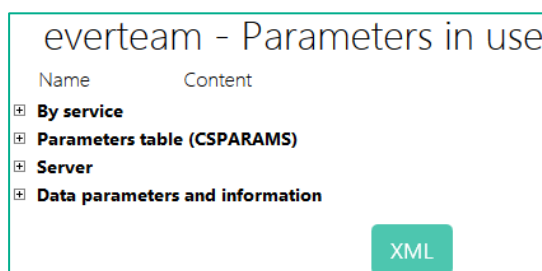
- [HTMLC](#) and [CONTROL](#): definition and description of the controls declared in table structure definitions.
- [HTML](#): definition of input and display fields (HTML objects) used in generic forms.
- [LANG](#): declaration of language substitutions. This parameter lets you substitute one language for another if everteam does not contain a translation for a particular language.
- [PARAM](#): declaration of parameters used by everteam's services.

2.1.7 Parameters in use

This menu displays read-only lists of parameters:

- [By service](#): list of parameters sorted by service, stored in memory.
- [Parameters table](#): list of the parameters stored in CSPARAMS.
- [Server](#): list of the parameters in the current setup. These parameters are not necessarily stored in CSPARAMS.
- [Data parameters and information](#): list of the parameters stored in memory, sorted by alphabetical order.

A click on [\[XML\]](#) exports the contents of these lists in XML format.





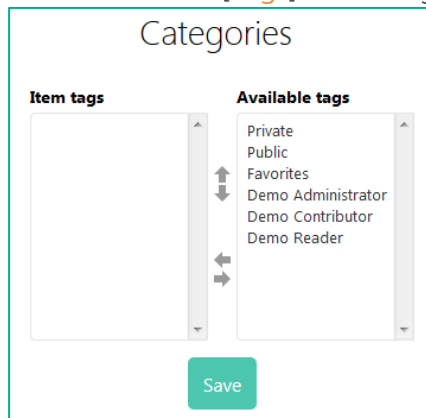
Parameters in use submenu

Access to this feature depends on a core security right: [CORE_PARAMS.GETALLPARAMS](#) (List all parameters). It can be managed in the [everteam.core > Security > Service security > everteam.core](#) menu.


2.2 Review of how parameters are saved

For each parameter – regardless of the menu used – are displayed:

- A  button [[Copy the XML parameter to the database](#)] to transfer the value in the external file ([param.xml](#)) to the [CSPARAMS](#) table. This icon is masked when the value in the XML file is identical to the one in the database (CSPARAMS).
- A  button [[Tags](#)] for adding extra information in the form of a tag (declared in [et.tag](#)).



Managing categories

- A  button [[Add a preference](#)] to declare preferences for a value for a user or a role (available for four UI parameters: [DocAutoLoad](#), [DocOpenWay](#), [viewersConfiguration](#), [ViewThumbnails](#)).
- A message in red highlights any inconsistencies between the parameters and the environment for the administrator, and gives help on correcting it.

Reference

For more information on managing parameters, please refer to the [CORE Parameters guide](#).

2.3 Core parameters

Some of these parameters are essential for all connections to everteam. They were declared when the application was installed.

Reference

Please refer to [everteam installation documentation](#).

2.3.1 Processing at connection time

These parameters are stored in memory when a connection is made from a browser.

The values displayed in this window come from either the [CSPARAMS](#) table (takes precedence), or from the [param.xml](#) file (`./apps/core/conf`). How the different values are used when connecting is shown in the table below. This depends on whether or not the parameter is in [CSPARAMS](#) and on the value stored there compared to the one declared in the [param.xml](#) file.

The value of each parameter, stored in memory when the connection is made and displayed in the **Core** menu, is shown in the column on the right in bold:

Case	Value stored in the CSPARAMS table	Value in the param.xml file	Value displayed on connection
1	Val 1	Val 1	Val 1
2	Val 1	Val 2	Val 1
3	Val 1		Val 1
4		Val 2	Val 2
5	Val 1	Val 2 if "priority=true" in param.xml	Val 2

Summary of how parameters are stored in memory when connecting

It is possible to force the value stored in **param.xml** by giving it the attribute **priority=true**.

For example to use the **core-contexturl** parameter to indicate that the server is running locally:

```
<param priority= "true">
  <PARAMSEQ>core-contexturl</PARAMSEQ>
  <USERCODE>all</USERCODE>
  <DESCRIPTION></DESCRIPTION>
  <CONTENT>http://myserver:8080/es510</CONTENT>
</param>
```

2.3.2 Core parameters list

Connection parameters

Reference

For more information on the core-contexturl parameter, please refer to the CORE Security guide.

Parameter Name	Definition	Default value
AlwaysForceLogin	Force login if the user already has another session open <i>true / false</i>	<i>false</i>
core-contexturl	everteam application server URL (mandatory)	
DocJAIPrefix	First part of the path to the ECM document storage server (with / or \ at the end)	

DocRoot	Second part of the path to the ECM document storage server (neither / nor \ at the end)	
DocRootPathBuilder	Configuration of the algorithm for calculating storage paths	
DocRootPrefix	URL to EDM document repository on the server (with / at the end). <i>Do not write</i> localhost	
DocESFtpPrefix	esftp URI driver for accessing an ftp server	
DocUseLocalURI	Use ESURI local: driver when storage path is empty <i>true / false</i>	<i>false</i>
ldapfactoryclass	JNDI context factory class name to access LDAP (context factory)	
mailer	Mail server connection parameters: SMTP address / From / To / Subject	
ModeUpdFile	Upload mode for attached documents <i>Upload, Applet, Japplet, DLL</i>	<i>Upload</i>
NumberOfFailedLogin	Number of authorized failed logins in <i>TimeOfFailedLogin</i> period	<i>3</i>
TimeOfFailedLogin	Number of hours preceding failed login to take into account when determining whether <i>NumberOfFailedLogin</i> has been exceeded. Default: 1	<i>1</i>

Security parameters

Reference

The security parameters concern rights on objects in the whole everteam application, regardless of database, table or records. For more information, please refer to the CORE Security guide.

Parameter Name	Definition	Default value
CheckFileFormat	Type of format check on deposited document and its authorization (depends on <i>FileFormatList</i>) <i>NO / SIGNAL / BLOCK</i>	<i>BLOCK</i>

CheckTableQuery	Check access rights on SQL queries <i>true / false</i>	<i>true</i>
CheckTables	Check table read access rights <i>true / false</i>	<i>true</i>
CheckTemplates	Check user's access to templates (class <code>clsParams</code>). <i>true / false</i>	<i>true</i>
ExecQuerySafe	Forbid dangerous SQL commands like INSERT, UPDATE ,DROP, etc. <i>true / false</i>	<i>true</i>
Sec_CheckOnRead	Check read access rights on record for <code>GetRecord</code> servlet <i>true / false</i>	<i>true</i>
Sec_CheckServlets	Authorize access to the servlets and their parameters, via regular expressions. The filter is stored in the <code>CSCHECKSERVLET</code> table. <i>true / false</i>	<i>true</i>
Sec_GlobalSystem	Manage the security policy as a "flat" list [<i>true</i>] or relationally [<i>false</i>].	<i>false</i>
Sec_MayModifyMenus	Permit to modify menus, via the menus API <i>true / false</i>	<i>true</i>
Sec_MayReloadTemplate Map	Permit to reload the TemplateMap when doing a Reload <i>true / false</i>	<i>true</i>
Sec_MayWriteDB	Permit to write in the database when loading the application <i>true / false</i>	<i>true</i>
Sec_MayWriteFS	Permit to write files to the file system or other devices <i>true/tmp</i> (only to the temporary directory)/ <i>false</i>	<i>true</i>

Sec_Record	Check update rights on a record after an SQL query (UPDATE and INSERT). <i>true / false</i>	<i>false</i>
Sec_ShowErrorMessage	Display error messages in plain text. <i>true / false</i>	<i>false</i>
SecurityLearningMode	Enable Security learning mode <i>true / false</i>	<i>false</i>

Other parameters

Parameter Name	Definition	Default value
AbbyyRS	Parameters for using ABBYY 'OCR in server mode	
AccessMode	Access mode for everteam records: <i>read-write</i> or <i>read</i>	<i>read-write</i>
AnnotDBName	Annotation database name	<i>dPortal</i>
CheckUnicityContentOn	Mode on which record content uniqueness is checked <i>ALL INSERT UPDATE</i>	<i>ALL</i>
ClassRepository	Repository of predefined indirections to everteam classes	
DBMaxColumnNameLength	Maximum column name length in everteam tables	<i>18</i>
DBMaxTableNameLength	Maximum table name length for everteam tables	<i>18</i>
DCOMProvider	Connections to DCOM services using uuids supplied by Ever Team	
Def_Encoding	Character encoding in everteam <i>ISO-8859-1 UTF8</i>	<i>ISO-8859-1</i>
DeleteDocs	Delete attached documents when records are deleted <i>true / false</i>	<i>false</i>
DFL_DetailedLogging	Enable DataFlow logs <i>yes / no</i>	<i>no</i>

DocSecurityPrefix	URL base for accessing Security services (CEFCBase)	
DocClipboardAsk	Check if a window is already open in copy and paste mode <i>true / false</i>	<i>false</i>
DocClipboardPersistent	Keep a copy of the document in memory <i>true / false</i>	<i>false</i>
DocCloudPrefix	Configuration for accessing cloud files using the cloud: driver	<i>XML stream</i>
DocESFtpPrefix	esftp URI driver for accessing an ftp server	<i>XML stream</i>
DocHCPrefix	Configuration for accessing HCP v2.6, v3 and v4 services via the hcp: uri driver	<i>XML stream</i>
DocLocalPrefix	Parameters to access local storage via the local: driver	
DocOSSPrefix	Information required in order to connect to OSS main server: protocol host:port/webapp database login password	
EscapeQuotes	Replace double quotes (") by (") <i>yes / no</i>	<i>no</i>
escmPrfAllViews	Content Management: use one or all available everteam views per profile <i>true</i> (all the views)/ <i>false</i> (one view)	<i>false</i>
escmSearch	Content Management: Search form used <i>SEARCH/SEARCHLIKE</i>	<i>SEARCH</i>
escmUsePRF	Content Management: Work with profiles <i>true / false</i>	<i>true</i>
ESMIG_DATABASE_MAPPING	Mapping between types of database supported and their drivers, for SQL	

	migration actions	
ESMIG_FORBIDDEN_COMMANDS	System commands forbidden in a migration	
FileFormatList	List of file formats authorized for document upload	<i>Flux XML</i>
GlobalSignature	Sign records and documents, and store a global signature <i>true / false</i>	<i>true</i>
IncomingURIEncoding	Encoding used by the web server to decode incoming http requests (ensures compatibility with the application server) <i>ISO-8859-1/UTF-8</i>	<i>ISO-8859-1</i>
SO_Encoding	Encoding for configuration files in the ./webapp/conf directory	<i>UTF-8</i>
IsUpper	Use UPPER on columns in SQL WHERE clauses <i>yes / no</i>	<i>yes</i>
KeysSplitterSliceSize	Number of pieces to split the key list in to in the query sent to the GetRecords servlet	<i>100</i>
LastModified	Compare last time the object URL was modified with the date of the file kept in cache <i>true / false</i>	<i>false</i>
LinkKeys	Manage reverse links between tables (CSLINKS table) <i>true / false</i>	<i>false</i>
Links_externalize	If <i>LinkKeys</i> = <i>yes</i> , CSLINKS table is not in the system database <i>yes / no</i>	<i>no</i>
LoginConfig	Redefine parameters for different everteam access methods	
maxConcurrentScheduler	Number of simultaneous tasks for	<i>5</i>

s	RUNNING scheduler (see schedulerDelay)	
MaxLastModified	Maximum number of URLs kept in the server cache (see LastModified)	1000
ModeLockManager	Lock manager mode SQL / HSH	HSH
NFZ42013	Comply with the NFZ 42-013 standard (no overwriting of attached documents, log configuration) true / false	false
NonFilteredRoleSupercedesFilters	If the user has a role with no filters (CSFILTER), the filters in his other roles are not applied either. yes / no	no
OutputBufferSize	Output buffer size, in number of characters	8000
ResourceCacheTime	Number of minutes resources are kept in cache. 0 disables the resource caching feature	10
RetryCountOnDeletionException	When deleting, SQLServer sometimes throws a deadlock exception. This parameter sets the number of times the deletion is retried.	1
RetryCountOnInsertionException	When inserting, SQLServer sometimes throws a deadlock exception. This parameter sets the number of times the insertion is retried.	1
RMI Services	Features accessible via RMI on a single port	
RoleAPI_CacheSize	Number of roles in cache	500
schedulerDelay	Delay (in ms) before re-running a scheduler's task when maxConcurrentSchedulers is reached.	
schedulemanager	Method(s) to execute at evertteam startup or at a scheduled time, and	

	then according to a given frequency (in ms)	
SendMsgMail	Send core messages by email <i>yes / no</i>	<i>no</i>
SessionCallStack	Store the call stack for the session	<i>false</i>
SessionInfo_UseldForCache	Creation and reuse of sessions on a remote everteam by the SessionInfo class. <i>false</i> : one remote session per identity <i>true</i> : one remote session per local session	<i>false</i>
TDescMetaFullInitialize	Load table metadata and description (<i>TableDesc</i>) into memory <i>true / false</i>	<i>false</i>
TemplateCache	Keep loaded templates in memory <i>yes / no</i>	<i>no</i>
ThumbnailsSmall	Display attached documents as icons or thumbnails <i>yes</i> (icon) / <i>no</i> (thumbnail)	<i>no</i>
TmpDir	Location of temporary files: TmpDirPath is a file system path, TmpDirURL is the URL pointing to the same directory	
TokenHandler	Name of a class to handle special identity tokens, not handled by the standard ESToken class	
TreeDynDatabase	Name of the database containing the et.FilingPlan (dynamic tree view) tables	<i>dportal</i>
triggermanager	Declaration of triggers active in the application for insert/update/delete	
UKVClass	Class to get unique key field value <i>ukvDefault/ukvSelect/ukvUpdSel</i>	<i>ukvSelect</i>
URICacheTime	Number of seconds URIS are kept in cache. <i>0</i> disables the caching feature.	<i>10</i>

URILabelsAttributesCount	Number of fields used to display URI labels (ESDB and GETDOCS)	3
userTableFilterExcludeRegExp	Tables to be removed from list: used by system to provide list of 'user' (i.e. NOT SYSTEM) tables. All tables declared in everteam are filtered using this RegExp. Ones which match are removed from the 'User table' list	
userTableFilterIncludeRegExp	Tables to be included in the 'user table' list (NOT SYSTEM tables). All tables declared in everteam are filtered using this RegExp. Ones which match are added to the 'User table' list	
VerifyDocAfterPut	Compare the content which has just been stored with the source content	false
wanticon	Show specific icons in each menu <i>yes / no</i>	yes
ZipEntryCharset	List of encoding formats to test for when detecting a zip's encoding	UTF-8,CP437

3 Management

The sub-menus in the **everteam.core > Management** menu are:

- **Refresh**: reload application parameters after they have been modified.
- **Dictionary**: manage a customized dictionary for everteam to use to translate the application, in addition to the default file, all.xml in the ..\conf\DicoKW folder of the webapp.
- **Forms**: manage and load into everteam all the forms (or templates) on which you wish to define specific rights.
- **Locks**: view locked records.
- **Do links**: rebuild the links between tables.

3.1 Refresh

It is essential to refresh the parameters when you have modified an application object after it was loaded into memory when the application server was started.

You may choose the affected objects from the list of all possible objects. If you check the first option, **All**, all the parameters are updated.

This refresh lets you continue to work in the same browser window, with the new values of the updated parameters.

Reload parameters

All

Params

Core

Tables

DataBases

Dico

TableForms

Other

TableDescs

Profiles

Index

Menus

TemplateMap

TemplateCache

Messages

Filters

Js

List of objects to refresh (extract)

When the reload is finished, the list of objects refreshed is displayed.

This action depends on a core security right: [CORE_MANAGEMENT.RELOAD](#) (Reload static objects). It can be managed in the [everteam.core > Security > Service security > everteam.core](#) menu.

3.2 Dictionary

3.2.1 Basics

The dictionary generated in everteam provides translation for the environment of the infrastructure and its applications. It is supplied in an external file ([all.xml](#)) in the \conf\DicoKWs directory of the webapp. This file is loaded into memory when the application server is started.

Each phrase is represented by a keyword which acts as a reference. It is in the [<KEYWORD>](#) tag.

Example: the record for the "AllFiles" keyword with its French, English, Spanish, German and Arabic translations:

```
<DicoKW>
  <APPLIC><![CDATA[es]]></APPLIC>
  <KEYWORD><![CDATA[AllFiles]]></KEYWORD>
  <EN><![CDATA[All]]></EN>
  <FR><![CDATA[Tous les fichiers]]></FR>
  <ES><![CDATA[Todos los ficheros]]></ES>
  <DE><![CDATA[Alle Dateien]]></DE>
  <IT />
  <SV />
  <NO />
  <NL />
  <AR><![CDATA[جميع الملفات]]></AR>
  <HE />
  <JA />
  <FI />
  <DA />
  <PT />
  <EL />
  <EU />
```

<CA />

</DicoKW>

"AllFiles" keyword with its French, English, Spanish, German and Arabic translations:

To add or manage new words or phrases specific to your application, use the [Management > Dictionary](#) menu. It gives access to the application dictionary table (CSNEWDICO) in the standard interface, with search and related operations (insert, update, delete, publish, etc.).

<div> <div>> **</div> <div>> Fields with no Group</div> <div>> System Information</div> </div>	
Key :	0
Key word :	mydatafile
Application :	es
EN English :	My data file
FR French :	Mon fichier de données
ES Spanish :	Mi fichero de datos
DE German :	Meine Datei

Phrase added to the application dictionary

3.2.2 Loading and priority

When the application server starts, the all.xml file is loaded into memory before the CSNEWDICO table. If there is a conflict on a [keyword \(KEYWORD\)](#), the CSNEWDICO table takes precedence. The duplicate in the all.xml file is erased from memory.

Reference

For more information on translating the interface, please refer to the [CORE Localization guide](#).

For more information on standard interface features, please refer to the [CORE Standard user interface guide](#).

3.3 Forms

3.3.1 Basics

Forms are created in the file system. They are used directly by everteam when required.

However, to have security on these forms in an application, for a role or some users, they must be known in the system database. They should therefore be loaded into the CSFORMS table. These forms can then be available when Security is being put in place.














The **Management > Forms** menu lists the forms stored in the **CSFORMS** table and lets you load the necessary forms.. Only the name and the description of the forms are loaded into the database, not the contents.

By default this table is empty. The results list is displayed with tools for viewing and updating depending on rights.

3.3.2 Loading

Via the menu **USED FORMS > Load forms**, everteam automatically loads all the forms in the *Templates* directories and related subdirectories, and JavaScript files. Afterwards it is possible to retain only the forms requiring security.

To load a batch of forms, click on the **[Submit]** button. When loading is complete, click on **[Back]** to see the list of forms.

No		▲ Name	▲ Description
1	<input type="checkbox"/>	 /scripts/Records/Record.js	../scripts/Records/Record.js
2	<input type="checkbox"/>	 /scripts/assistants/Assistants.js	../scripts/assistants/Assistants.js
3	<input type="checkbox"/>	 /scripts/assistants/genUtils.js	../scripts/assistants/genUtils.js
4	<input type="checkbox"/>	 /Templates/Password.htm	Password.htm
5	<input type="checkbox"/>	 /scripts/security.js	/scripts/security.js
6	<input type="checkbox"/>	 /scripts/pschecker/js/pschecker.js	/scripts/pschecker/js/pschecker.js
7	<input type="checkbox"/>	 /apps/core/templates/frameset/version.htm	/apps/core/templates/frameset/version.htm
8	<input type="checkbox"/>	 /Templates/default.htm	EverSuite - Connexion
9	<input type="checkbox"/>	 /scripts/TTFunctions.js	../scripts/TTFunctions.js
10	<input type="checkbox"/>	 /scripts/xmlextras.js	../scripts/xmlextras.js
11	<input type="checkbox"/>	 /apps/core/templates/frameset/index.html	/apps/core/templates/frameset/index.html
12	<input type="checkbox"/>	 /apps/ihtm/templates/checkConnexion.txt	/apps/ihtm/templates/checkConnexion.txt
13	<input type="checkbox"/>	 /apps/core/templates/frameset/index_help.html	\${tr.EverSuite} - \${tr.Menus}

Forms list (extract)

Select all the forms by checking the box at the top of the list. Deselect the forms you wish to retain.

Click on the **EDIT > Delete** menu to delete the unsecured forms.

Information

Deleted forms are not deleted from the ./Templates directory and remain operational.

Information

The USED FORMS > Update forms (v.4.2.2_04)] menu is to be used after migrating from version 422_04 or 510_01 to everteam 510_02.

3.4 Locks



The **Management > Locks** menu lets the administrator view a snapshot of locked records (for example records currently being updated).

everteam locks records while they are being modified, to prevent simultaneous updates of the same data.

When a record goes into 'Edit' mode, it is locked automatically for a given period for the current user.


If another user tries to access a record for update, he will get an error message: 'Locked Record'

In the **CSPARAMS** table, the **DB_MaxLockDuration** parameter fixes the maximum time a record can be locked (in seconds).

Del	Database name	Table name	Record key	User	Date and Time
	dPortal	CSNEWDICO	0	admin	2013-10-14 11:20:37
	dPortal	ESDMS_DEMO	42	admin	2013-10-14 11:19:53

List of locked objects

A click on  refreshes the page.

A click on  unlocks all the records, after confirmation of the message: *Are you sure you want to clear the locks?*

A click on  unlocks the current record.

See

The LockManager class, for managing locks, is described in an Appendix

3.5 Do links














3.5.1 Basic concepts

The link from a main table to another table is put in place using the **link** control defined for a field. It is taken into account when inserting and updating. This field then contains the key of the record in the linked table.

The links between the two tables are automatically recorded in the link table, **CSLINKS**, if three conditions are met:

- the main table counter is handled by everteam.
- the control is a link or a tm
- the source field is the primary key of the linked table (ReturnField=DKEY).

There is one **CSLINKS** system table for all the everteam applications. It contains four fields for storing the link between two tables: the keys of the main table and the linked table, as well as the keys of the records in these two tables.

No		▲ Table key	▲ Record key	▲ Related table key	▲ Related record key
1		 744	1	729	8
2		 744	2	729	9
3		 744	3	729	8
4		 828	0	829	0
5		 862	0	856	-1
6		 862	0	862	-1

CSLINKS table

The key of a table is the key in the CSTABLES table.

3.5.2 Execution

Initializing the links consists of destroying the CSLINKS table and rebuilding it with information on links in every application table whose counter is managed by everteam and which contains one or more links. everteam asks for confirmation before starting the process: *Do you wish to initialize the links?*

4 Technical administration

The sub-menus in the [everteam.core > Technical administration](#) menu are:

- Evertteam variables
- Token creation
- Thread pools

4.1 evertteam variables

The [everteam.core > Technical administration > evertteam variables](#) menu gives access to evertteam variables, arranged in three blocks:

- Static data
- Global variables
- Session parameters

You just enter the name (or part of the name) of the variable you are looking for (with no truncation character) to see the results displayed in these three blocks:

Enter a variable name (Case sensitive)

Search

or press Ctrl + F (after expanding the blocks of variables)

Reinitialize

Static data
EP_LinkControlProcedure_
EverLinked_DBNAME
LinkKeys
Links_Externalize

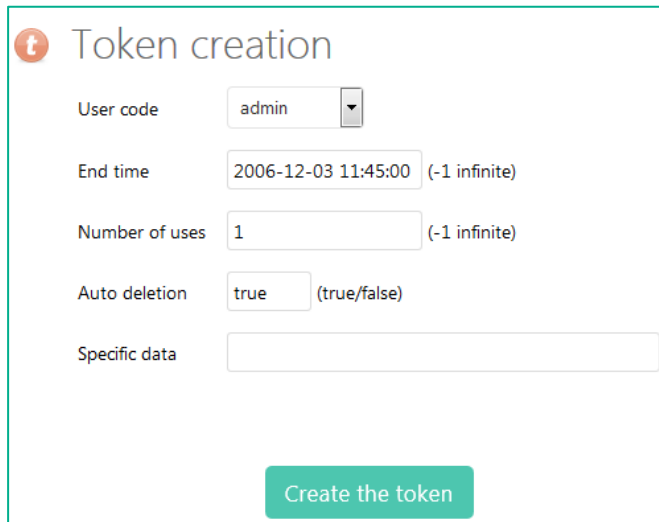
Global variables
No hits

Session parameters
LinkType

Finding variables

It is possible to use the [Ctrl] + [F] keys as long as the blocks have been expanded first.

4.2 Token creation



The image shows a 'Token creation' form with a teal border. At the top left is a teal circle with a white 't'. The title 'Token creation' is in a large, dark font. Below the title are five fields: 'User code' with a dropdown menu showing 'admin'; 'End time' with a date/time picker showing '2006-12-03 11:45:00' and a link to '(-1 infinite)'; 'Number of uses' with a text input showing '1' and a link to '(-1 infinite)'; 'Auto deletion' with a checkbox labeled 'true' and a link to '(true/false)'; and 'Specific data' with a large empty text area. At the bottom center is a teal button with the text 'Create the token'.

Token creation form

Reference

Please refer to the **CORE Authentication tokens guide**.

4.3 Thread pools

This management menu lets an everteam application administrator monitor the total number of everteam *threads*, in other words the number of jobs being executed at the same time. It shows execution time and memory usage.

Threads are monitored by service and by job family.

The administrator may configure *ThreadPools* to meet particular requirements for priority or load.

With this organization, the administrator can:

- Follow the progress of current jobs
- produce statistics
- produce information on thread pool activity.

The menu has four tabs:

- Job families
- Execution groups
- Mappings
- Monitor

4.3.1 Basic concepts

The idea of jobs to be done must be distinguished from the *threads* which execute them. Generic *threads* can execute any jobs; they are polyvalent and take on whatever tasks they are allocated. They may be able to carry out parallel tasks, depending on the operating system and the type of machine.

A limited number of inactive *threads* are available in a pool. When a job request comes in, a thread is taken from the pool and assigned to this job, then it returns when the job is complete.

If too many requests arrive and the jobs take too long, the *thread pool* may be empty and a queue of pending jobs is created. In this case:

- either the job requestor waits until one of these threads is free and does the job,
- or the requestor goes away to do something else and comes back later to see if a thread has become available.

To optimize the Java virtual machine, the number of threads must be limited. A fixed total of threads must be distributed such that the maximum number of jobs are accepted and that as far as possible they run without hindering each other. It was decided to work with small pools per module.

4.3.2 ThreadPools

The everteam *ThreadPool* uses a basic Sun *ThreadPool* for each execution group.

- One with an unlimited size for the everteam core: "system" job family
- Others with a fixed size for the rest: "application" job families

Their definition and configuration is persistent and can be modified in the administration module, where others can also be created for private use.

4.3.3 Job families

This tab lists the job families which already exist and gives the administrator the possibility of creating new families.

Families may be declared by a developer or by an everteam application administrator.

Job families

ServiceInit

Service_core

SystemDefault

Enter name for new job family


Enter name for new job family

Job families

Create a family



Give the family a name and click on the  icon. The list updates automatically:

Enter name for new job family

sis 

Creating an SIS job family


4.3.4 Execution groups

Name	Max threads	Priority	
Default	10	5	
SystemDefault	-1	5	

New execution group name

Max thread value

Execution group priority



Execution groups

4.3.5 Mappings

Cancel Save

Available job families

Service_core

Execution groups

Default


SystemDefault

ServiceInit


SystemDefault

Mappings

4.3.6 Monitor



Execution group

Type 

Name SystemDefault

Identifier -1

Max threads 10

Priority 5

Thread Pool

Keep alive time	60
Core size / Max size	<div><div></div></div> 0/2147483647
Current size / Largest size	<div><div></div></div> 0/4
Active tasks	0
Suspended tasks	0
Completed / Scheduled	<div><div></div></div> 4/4

Monitor

5 Monitoring

5.1 Monitor

The [everteam.core > Monitoring > Monitor](#) menu is for monitoring the server. It analyzes:

- current connections
- tables to be indexed. These are the tables which have been modified, without being indexed automatically (incremental indexing)
- servlets being executed
- memory free compared to virtual memory declared in Java.

Access to this feature depends on a core security right: [CORE_MANAGEMENT.MONITOR](#) (Monitoring). It can be managed in the [everteam.core > Security > Service security > everteam.core](#) menu.

Connections	License	Interactive sessions
License owner - PRODOC		
Connections		
Database name	Free connections	Used connections
2 : dPortal	5	0
1 : LesEchos	0	0
<div>Restart all connections</div> <div>Show busy connection callStack</div>		

Analysis of connections

The [[Restart all connections](#)] button reinitializes all connections, that is, the system keeps only the active connections corresponding to an open browser window.

The [[Show busy connection callStack](#)] button displays logs of calls which opened a connection to a database.

Memory		
Totale	Libre	Utilisée
946864128	383374768	563489360

Memory usage

Connections

License

Interactive sessions

EverSuite sessions



As	Mode	Type	User	Identifier	Date	WebApp
1	1	ESessionHttp	admin	Confidential	2016-08-19 17:42:27	Confidential

everteam sessions

5.2 Scheduler

The [everteam.core > Monitoring > Scheduler](#) menu lists the schedulers and their status.

Each scheduler can be started () or stopped ()

Scheduler					
1			ExecuteTimers.callFromScheduler	ExecuteTimers.callFromScheduler	Status : Active
2		PeriodicIndexRoll	PeriodicIndexRoll	PeriodicIndexRoll	Status : Active
					10000 Manual start
					cron:0 01 00 * * ? Manual start

List of Scheduler instances

These actions depend on some core security rights: [CORE_SCHEDULER.LIST](#) (List schedulers) and [CORE_SCHEDULER.START_STOP](#) (Start and stop schedulers). These can be managed in the [everteam.core > Security > Service security > everteam.core](#) menu.

5.3 Update counters

5.3.1 Basic concepts

This affects application tables whose counter is handled by everteam. The [everteam counter](#) option must be checked in the table structure definition when it is declared in everteam. For the other tables, the counter is handled directly by the RDBMS.

The counters are stored in the [CSCOUNT](#) system table. Updating the counters involves finding - for each table - the maximum key (last record) and resetting the counter to this value.

Other specific counters may be created for a particular service.

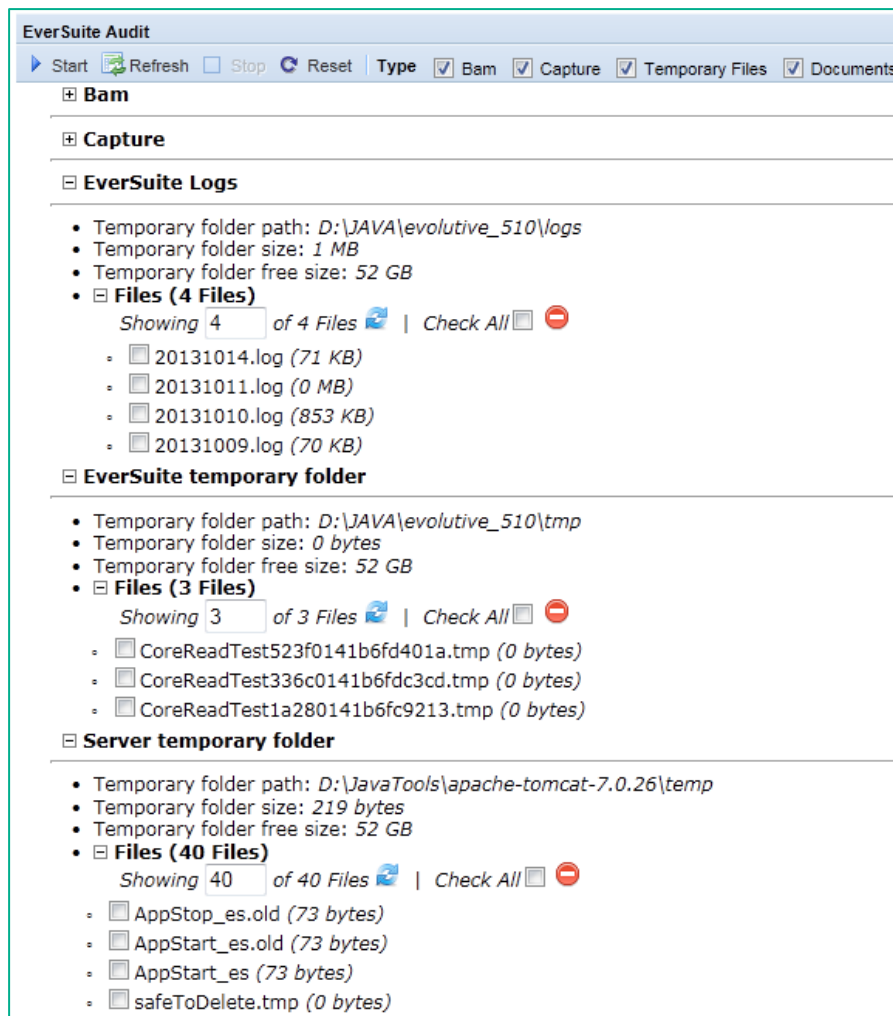
5.3.2 Execution

Click on the [everteam.core > Monitoring > Update counters](#) menu. everteam asks you to confirm by clicking [[Submit](#)], before running the process.

When processing is complete, the following message is displayed: *Count update OK*.

5.4 Audit

The [everteam.core > Monitoring > Audit](#) menu lets you run an audit of the application, by selecting the type(s) of audit then clicking on [[Start](#)]:



Audit extract

This action depends on a core security right: [CORE_GENERAL.AUDIT](#) (Application audit). It can be managed in the [everteam.core > Security > Service security > everteam.core](#) menu.

Results can only be consulted on the screen.

5.4.1 Activity statistics

This audit shows the number of records in the table containing logs for the Activity statistics service (et.bis), with dates for the oldest and the most recent entries.

5.4.2 Data import

The audit of the Data import service calculates job statistics.

Data import
<ul style="list-style-type: none"> Total Number: 87 Valid jobs: 6 Running jobs: 1 Error jobs: 80 Date of first job: 2016-06-08 Date of last job: 2016-07-19

Audit example

5.4.3 Temporary files

Several directories are analyzed and it is possible to delete files selected from the list. For each directory, its path, size and available space are displayed. The size of each file is also given.

Everteam log files

Two types of file are shown:

- everteam *ESCMD* logs. If a path has not been defined in the *Log_Path* parameter, these files are stored in the logs directory of the everteam webapp.

For detailed information on the Log_Path parameter, please refer to the CORE Parameters guide.

- log4j logs. These are logs defined by *appenders* in the webapp's log4j.xml file.

everteam temporary folder

This section displays the contents of the temporary directory given in the *TmpDir* core parameter, for example *D:\webapps\everteam\tmp*

Server temporary folder

This section lists the files in the application server temporary directory, for example */home/local/tomcat-7.0.55/temp*.

5.4.4 Documents

Two types of inconsistency can be picked up by the document audit:

- Unreferenced documents

This lists all addresses of documents stored by everteam without being referenced in a table.

Caution

This analysis is only possible when document storage is based on the DocRoot, DocJAIPrefix and DocJAIPrefix parameters (DocUseLocalURI parameter set to false). However, for security reasons, we strongly recommend that you do not use this type of storage.

Please refer to the CORE Security guide for information on how to implement secure storage.

- Non-existent documents

This is the list of URIs referenced in everteam tables but whose path does not exist.

6 Appendices

6.1 LockManager class

The LockManager class is responsible for managing locks.

It is a "singleton", which means that it is impossible to instantiate it directly using `new`. Instead you must use the following syntax:

```
LockManager lm = LockManager.getInstance();
```

It contains the following methods:

- `lock`: this method must be called when a record is read for update. It stores the references of the record to be edited. If another user tries to access the same record for update, he will get a warning message. Once the maximum time allowed for a lock is past, the record will be automatically unlocked if it is required by another user.
- `free`: unlock a blocked record.
- `beginTransaction`: indicate to the LockManager that a bulk transaction is starting. Any unlock request received during the transaction is queued pending the "commit" action.
- `commit`: unlock the records blocked by the transaction.
- `rollback`: keep the records blocked.

Two other methods permit the administrator to manage the contents of this class:

- `list`: prepare the list of blocked record references in XML format
- `clear`: initialize the list of blocked references.