

Caché 数据库 编程规范 V1.15

东华医为科技有限公司

2021 年 03 月 25 日

版本历史

版本	作者	参与者	发布日期	修改摘要
1.00	杨艳秀		2008-05-04	初始版本
1.10	杨艳秀	苏明亮、吴群科、郭宗涛、马利华、周帆	2008-08-14	整体内容修改
1.11	杨艳秀	苏明亮、郭宗涛、吴群科、周志强、周帆、胡华明、刘书凤	2008-08-28	整体内容修改
1.12	杨艳秀	符德东	2017-08-09	zn 及 \$namespace 的使用规范(新增)
				修改临时 global 命名规范
1.13	杨艳秀	符德东、汪会财	2017-12-18	ProcedurtBlock 使用规范 (5.3)
1.14	杨艳秀	周志强、符德东、汪会财	2019-06-21	修改 Global 命名及类命名规范
1.15	汪会财	技术委员会 Caché小组	2021-02-27	实体类的 Storage 存储类型及计数器说明, 业务逻辑类命名, 删除点式程序示例, 事务示例修改, 增加领域附表, 增加 csp 命名与书写规范

Copyright © 2021 DHC MediWay Technology Co., Ltd. All rights reserved

请不要给第三方传阅

目 录

1. 编写目的.....	1
2. 命名规则.....	1
2.1. 基本原则.....	1
2.2. 函数（方法）命名.....	1
2.3. 变量命名.....	1
2.4. 实体类、Global 命名规则及定义.....	3
2.5. 业务逻辑类命名.....	4
2.6. CSP 的命名及规范.....	4
3. 注释规则.....	5
3.1. 基本原则.....	5
3.2. Routine、类注释.....	5
3.3. 函数、类方法注释.....	5
3.4. 代码注释.....	6
3.5. 组件 Message.....	6
4. 版式规则.....	6
5. 其他规范.....	7
5.1. Caché数据库中事务的应用.....	7
5.2. Caché数据库临时 GLobal 的命名规范.....	8
5.3. 类及 M 中 ProcedureBlock 属性及 New 的使用规范.....	8
5.4. 关于 zn 及\$namespace 的使用规范.....	8
6. 表的扩展.....	9
7. 代码检查.....	9
8. 代码优化.....	10

1. 编写目的

该文档目的在于规范 Caché 数据库的程序开发，以方便开发人员编写程序。

2. 命名规则

命名规则规定了程序中各种标识符的命名原则。包括：M Routine 命名，类命名，函数名，变量命名、GLOBAL 命名。

2.1. 基本原则

- 标识符应采用英文单词、英文单词缩写或其组合，用词应当准确，可读性要好。一般不要使用形如 i、xxx、temp 等无意义的名称。
- 在保证可读性的前提下，尽量缩短标识符的长度。通过减少标识符中单词的数量可以有效地缩短长度。但不要为了缩短长度而使用生僻或不规范的缩写。

2.2. 函数（方法）命名

- 推荐使用动宾结构，函数名应清晰反应函数的用途和功能。
- 当名称由多个单词组成时，单词首字母大写。
- 函数名最长不超过 30 个字符。

正确示例	错误示例
SetValue()	setvalue() //单词首字母没有大写
SearchPat()	Sousuo() //不应用拼音描述 Search() //非动宾结构
SetInsertVars()	TypeAdd() //非动宾结构

2.3. 变量命名

- 变量命名一般应使用“名词”或者“形容词+名词”。有时为了更醒目地表示变量之间的相关性，也可以采用“名词+形容词”的命名方式。
- 变量命名要直观且可以拼读，可望文知义，采用英文单词或组合，便于记忆和阅读，切忌使用汉语拼音来命名，程序中的英文单词一般不会太复杂，用词应当准确。
- 当名称由多个单词组成时，名称首字母可以大写或小写，之后的单词首字母大写。

- 避免出现仅靠大小写区分的相似的标识符，如 x，X。
- 避免出现单个字母的命名，如 a，b，c 等。
- 尽量避免名字中出现数字编号 value1，value2。
- 对于计数的循环变量可以使用 i, j, k, l, m, n。
- 对于在 HIS 系统中有固定意义的变量、参数、组件元素，命名必须按照规定的名称命名。见 《固定意义变量、参数、组件元素命名列表》。
- 对于在 HIS 系统中有固定意义的、但使用《固定意义变量、参数、组件元素命名列表》不能唯一表示的其他变量，命名必须按照规定的规则命名。见《固定含义词根列表》。

[前缀]+规定词根+[后缀]、规定词根 1+规定词根 2

固定意义变量、参数、组件元素命名列表

意义	规定名称
就诊记录指针	EpisodeID
患者指针，病人内部唯一标识	PatientID
患者病历指针，和就诊记录指针对应，但不等于它，用来确定诊断指针	mradm
医嘱指针	OEOrdItemID
医嘱状态	OEORItemStatus
医嘱名称	ARCIMDesc
病区指针	WardID

固定含义词根列表

变量意义	规定词根	变量举例
就诊记录指针	PAADM,ADM,Episode	curEpisodeID
患者指针	PAPER,PAPMI	PAPERName1,PAPERName2
病人医嘱指针	OEORI	OEORISttDate
医嘱	ARCIM	
病案记录	MRADM	
科室/部门	CTLOC,DEPT	
医生	CP,CTPCP	
预约	APPT	

2.4. 实体类、Global 命名规则及定义

对于各产品线所使用的基础数据、配置数据及业务数据存储的表/实体类、Global 遵循以下命名规范：

1) 基础数据表/实体类、Global

表命名	Global 命名
CT.领域编码(1).产品线.功能名称(2)	^CT.领域编码(1).产品线.功能名称(2)

2) 配置数据表/实体类、Global

表命名	Global 命名
CF.领域编码(1).产品线.功能名称(2)	^CF.领域编码(1).产品线.功能名称(2)

3) 业务数据表/实体类、Global

表命名	Global 命名
领域编码(1).产品线.功能名称(2)	^领域编码(1).产品线.功能名称(2)

其中(1) 为产品线归属的领域编码；(2)功能命名要求开头字母大写，单词应清晰表达“表和 Global”的用途和功能，当名称由多个单词构成时，单词的首字母要大写。

4) 实体类定义规范

(1) 类要求继承 User.Abstract

如：Class User.SSUser Extends (%Persistent, User.Abstract){
}

(2) 表/实体类的 Storage 存储类型及计数器：

- 表/实体类 Global 数据存储类型使用 CacheStorage 类型，不再使用 CacheSQLStorage 类型；
- 表/实体类的 Master Map 主存储为类名后面加 D、索引存储类名后面加 I、流存储类名后面加 S；父子表的可以考虑使用同一存储加节点 ({%%PARENT}, ”子节点”)；
- 计数器不再使用 ^mdata(“表名字”)，可以使用 \$i(表的主数据存储节点)，子表的计数器要特别注意与父表对应记录之间的增长关系（每条子记录是否都是从 1 开始顺序增加）；

- 字段/属性的 Collation 要与对应的 Index 的 Collation 保持一致。

2.5. 业务逻辑类命名

业务逻辑类名称
领域编码(1).产品线.分层编码.功能名称(2)

其中(1) 为产品线归属的领域编码；(2)功能命名使用英文单词（缩写）组成，单词应清晰表达业务逻辑类的用途和功能，当名称由多个单词构成时，单词的首字母要大写。

其中，分层编码内容如下表所示：

分层编码	备注
BL	产品线的业务逻辑层。类要求继承 BSP.SYS.BL.Base
DTO	产品线的数据传输对象层。类要求继承 BSP.SYS.DTO.Base
COM	产品线的公用程序层。
SRV	产品线的对外部系统的服务层。

2.6. CSP 的命名及规范

1) CSP 的命名

CSP 的名称
领域编码(1).产品线.功能名称(2).csp

其中(1)为产品线归属的领域编码；(2)功能命名使用英文单词（缩写）组成，单词应清晰表达业务逻辑的用途和功能，当名称由多个单词构成时。全部采用小写格式

2) 所有 CSP 文件中必须包括以下内容

```
<csp:method name=OnPreHTTP arguments="" returntype=%Boolean>
    If ##Class (websys.SessionEvents) .SessionExpired()
quit 1
    quit 1
</csp:method>
```

3. 注释规则

3.1. 基本原则

注释的语言采用中文。

3.2. Routine、类注释

文件注释放在每个代码文件的开头，用于说明本文件的功能、作者、修改历史等相关信息。文件注释的格式举例如下：

///名称：

///描述：

///编写者：

///编写日期：

///产品组：

3.3. 函数、类方法注释

函数注释放在每个函数或类方法声明的上方，不要放在函数体或类方法的内部。用于说明该函数或方法的功能、出入口参数、调用注意事项等。注释以三条斜线（///）作为注释符，函数注释的格式举例如下：

///Creator: 作者

///CreatDate: 编写日期

///Description: 函数功能、性能等的描述

///Table: 被访问的表（此项仅对于牵扯到数据库操作的程序

///Input: 输入参数说明，包括每个参数的作用、取值说明及参数间关系。

///Output: 对输出参数的说明

///Return: 函数返回值的说明

///Others: 其它说明

3.4. 代码注释

即使采用可读性好的命名规则，注意程序书写的版式和逻辑结构的清晰，程序中还是会有一些地方需要用自然语言对代码进行注释。使用代码注释时需要注意以下几点：

- 虽然注释有助于理解代码，但注意不可过多地使用注释。注释是对代码的“提示”，而不是文档。程序中的注释不可喧宾夺主，注释太多了会让人眼花缭乱。
- 注释应当准确、易懂，防止注释有二义性。错误的注释不但无益反而有害。
- 注释以两条斜线（//）作为注释符。
- 边写代码边注释，修改代码同时修改相应的注释，以保证注释与代码的一致性。不再有用的注释要删除。
- 注释的位置应与被描述的代码相邻，可以放在代码的上方或右方，不可放在下方。
- 修改注释

在程序调试和测试阶段，如果需要修改较为稳定的代码，特别是修改别人编写的代码，必须进行修改注释。因为这种修改一般比较仓促，容易出错。一旦出现问题，修改注释可以帮助我们查找原因。在形成稳定版本时，需要将修改注释去除。修改注释的方式举例如下：

```
code added line 1; // +<修改日期> <修改者> <修改原因>
```

```
code added line 2; // +
```

```
...
```

```
code added line n; // +
```

3.5. 组件 Message

组件中 **Message** 命名参照变量命名规则，不能用数字或单个字母命名。

4. 版式规则

代码编写要整齐，for 语句，if 语句，以 4 个字符位作为缩进长度

正确示例	错误示例
<pre>If () { //..... } Else { //..... }</pre>	<pre>If () { //..... } Else { //..... }</pre>
Classmethod, 函数体以 tab 缩进	
<pre>ClassMethod Test() { Quit 0 //以 tab 缩进 }</pre>	

5. 其他规范

5.1. Caché数据库中事务的应用

Caché允许事务的嵌套，事务回滚时一直回滚到最外层事务。一个事务的起始与提交语句必须对应，因此要在一个函数或类方法内定义一个完整事务，并且在这个事务内一定要定义缺省的错误处理陷阱程序，错误处理陷阱程序要包含事务回滚命令。不能把事务的起始语句定义在一个函数内，而提交语句在另一个函数内。

```
Trans    ;
....
Set $ZT="ERROR"           //定义错误处理函数
TSTART                     //开始一个事务
....
If SQLCODE'=0 TROLLBACK   //根据结果主动回滚事务
....
If $TLevel>0 TCOMMIT      //提交事务
....
Quit ReturnValue          //正常结束
ERROR
Set ErrorMsg=$ZE          //得到系统返回的错误消息
If $TLevel>0 TROLLBACK    //如果存在事务，回滚事务
Set $ZT=""                //去除错误方法关联，此句必须写
Lock                      //解去当前进程内所有锁
Quit "<ERROR>"_ErrorMsg   //返回错误消息
```

5.2. Caché数据库临时 Global 的命名规范

对于临时 Global，命名规范为^TEMP*、^Temp*、^temp*、^TMP*、^Tmp*、^tmp*、^CacheTemp*，不允许有其它的命名。

DHC-TEMP 库文件不能做 Journal 记录，否则也会产生大量无用 journal 文件。

程序生成临时 global，处理完业务逻辑后，一定要在程序中 kill 掉，否则会造成 DHC-TEMP 库文件日益增大，占用磁盘空间。

5.3. 类及 M 中 ProcedureBlock 属性及 New 的使用规范

为了保证类方法和 M 函数中的变量只在当前的局部作用域内有效，避免跨类方法或 M 函数调用时重名变量不同作用域间的相互影响，导致结果错误，规定如下：

- 1) 所有新建类文件原则上必须使用默认的 ProcedureBlock 属性（cache5.2、cache2010、cache2016 下创建的类自动默认为 ProcedureBlock 属性）；
- 2) 如因使用特殊字符(如：X, @)等需要将类文件定义为 Not ProcedureBlock 属性时，则必须在类方法的起始位置使用 New(variable)语句定义所有变量；
- 3) M 文件中，M 函数的起始位置必须用 New(variable)语句定义所有变量。

5.4. 关于 zn 及 \$namespace 的使用规范

在 Caché数据库的程序开发中涉及切换 namespace 时，存在两种情况：

- 1) 永久地切换到另外一个 namespace 切并不再切换回原 namespace，则要使用 zn，因为 zn 提供更多 security 方面的检查。
- 2) 暂时切换到另外一个 namespace，执行完一些任务后还需要切换回原 namespace，则要使用 Set \$namespace = "ns"。因为 \$namespace 允许使用 NEW \$NAMESPACE，这样可以保留当前 namespace 的 stack；如果从新 namespace 的 routine 中 quit 或者运行中发生错误，那么会退到前一个 namespace 的 stack；在新的 namespace 中产生的 object 实例，Caché也会在切换到原来的 namespace 前关闭。示例代码如下：

```
WRITE "before: ", $NAMESPACE,!
DO Test
WRITE "after: ", $NAMESPACE,!
;Routine code
.....
QUIT
Test
NEW $NAMESPACE
SET $NAMESPACE="USER"
WRITE "testing: ", $NAMESPACE,!
; routine code
.....
QUIT
```

注：如果是运行 SQL、Object 等代码，Cache 都推荐运行在同一个 namespace 中。在切换 namespace 的时候如果有打开的 object 以及 SQL cursor 等，可能导致代码运行出现错误。因为切换 namespace 时数据库系统需要比较高的计算量，因此在应用代码中应该尽量避免切换 namespace，无论是 zn 还是 set \$namespace。

参考：

http://localhost:57772/csp/docbook/DocBook.UI.Page.cls?KEY=RCOS_vnamespace#RCOS_B53047

6. 表的扩展

对 TrakCare 原有类或表的扩展，类名或表名在原来的名字前面加 “DHC”。例如对原有的 OE_OrdItem 扩展，扩展后的表名为 DHC_OE_OrdItem。类名为 User.DHCOEOrdItem。存储 GLOBAL 可在原来的 GLOBAL 下标前增加字节点 “DHC”。例如，原来的为 ^OEORD(Ord,“I”,ChI)，扩展部分的 GLOBAL 为 ^OEORD(Ord,“I”,ChI,“DHC”)。

7. 代码检查

通过代码走读或审查方式对代码进行检查，代码走读主要是对程序的编程风格，如注释，命名等以及编程时易出错的内容进行检查，可由开发人员自己或开发人员交叉的方式进行。代码审查主要是对程序实现的功能及程序的稳定性，安全性，可靠性等进行

检查及评审，可通过自审，交叉审核或指定部门抽查等方式进行。

在使用新的开发规范后，首先由各产品主管负责自己产品组的代码检查工作，培训部也将定期进行代码抽查。

8. 代码优化

代码优化方面，不要一味的追求程序的效率，应当在满足正确性，可靠性，健壮性，可读等质量因素的前提下，设法提高程序的效率；以提高程序的全局效率为主，提高局部效率为辅；在优化程序的效率时，应当先找出限制效率的瓶颈，不要在无关紧要之处优化；先优化数据结构和算法，再优化执行代码；如果原有的代码质量比较好，尽量复用，但是不要修补很差劲的代码，应当重新编写。

按业务划分多个领域表

序号	领域编码	领域划分说明	备注
1	AUX	辅助类服务	陪送、配送
2	BDP	基础数据管理	
3	BILL	收费	
4	BSP	CA 接口、基础平台	
5	CIS	RIS、检验、病理、重症监护、手麻、血液净化、心电、影像系统	
6	CKB	知识库	
7	CONS	会诊	待定
8	CR	科研类	
9	CRM	随访	待定
10	DOC	医生	医生工作站
11	DW	数据仓库、综合查询	
12	EMR	病历	
13	HRP	物资、药库、设备、绩效奖金、财务管理、人事管理	
14	INSU	医保	
15	MA	医务管理、病案、医疗质量管理；流程管理	临床路径；归档、病案；不良事件、首页质控、电子病历质控；流程管理
16	NUR	临床护士、护理病历、护理管理	
17	PA	患者主索引	患者主索引、卡管理、账户管理
18	PAADM	患者就诊	挂号、出入转、分诊、叫号、床位
19	PE	体检	待定
20	PHA	药事管理	
21	PORTAL	门户	
22	RB	资源	资源、预约、排班
23	CDR	数据中心	
24	HIP	Hospital information platform	