

Debugger

Debugger

MPS provides an API for creating custom debuggers as well as integrating with debugger for java. See [Debugger features overview](#) for a description of MPS debugger features.

- [Integration with java debugger](#)
 - [Nodes to trace and breakpoints](#)
 - [Startup of a run configuration under java debugger](#)
 - [Custom viewers](#)
- [Creating a non-java debugger](#)
 - [About DebugInfoInitializer](#)
 - [.debug files](#)
 - [How to write DebugInfoInitializer](#)

Integration with java debugger

To integrate your java-generated language with java debugger, provided by MPS, you should specify:

- on which nodes breakpoints could be created;
- nodes which should be traced;
- how to start your application under debug;
- custom viewers for your data.

Not all of those steps are absolutely necessary; which are – depends on the language. See next parts for details.

Nodes to trace and breakpoints



This part is different in MPS2.0 See the changes in [MPS 2.0 documentation](#).

Suppose you have a language, let's call it `highLevelLanguage` which generates code on some `lowLevelLanguage`, which in turn is generated directly into text (there can be several other languages between `highLevelLanguage` and `lowLevelLanguage`, it does not really metter). Suppose that the text generated from `lowLevelLanguage` is essentially java, and you want to have your `highLevelLanguage` integrated with java debugger. See the following table:

	<code>lowLevelLanguage</code> is <code>baseLanguage</code>	<code>lowLevelLanguage</code> is not <code>baseLanguage</code>
<code>highLevelLanguage</code> extends <code>baseLanguage</code> (uses concepts <code>Statement</code> , <code>Expression</code> , <code>BaseMethodDeclaration</code> etc)	Do not have to do anything.	Fully implement <code>DebugInfoInitializer</code> for <code>lowLevelLanguage</code> .
<code>highLevelLanguage</code> does not extend <code>baseLanguage</code>	Specify breakpointable concepts in <code>DebugInfoInitializer</code> for <code>highLevelLanguage</code> .	Fully implement <code>DebugInfoInitializer</code> for <code>lowLevelLanguage</code> . Specify breakpointable concepts in <code>DebugInfoInitializer</code> for <code>highLevelLanguage</code> .

See section [About DebugInfoInitializer](#) of this document for further information.

Startup of a run configuration under java debugger

MPS provides a special language for creating run configurations for languages generated into java – `jetbrains.mps.baseLanguage.runConfigurations`. Those run configurations are able to start under debugger automatically. See [Run configurations for languages generated into java](#) for details.

Custom viewers



Note that in MPS2.0 M1 `customViewers` language has been significantly improved (see [MPS 2.0 documentation](#)).

When one views variables and fields in a variable view, one may want to define one's own way to show certain values. For instance, collections could be shown as a collection of elements rather than as an ordinary object with all its internal structure.

For creating custom viewers MPS has `jetbrains.mps.debug.customViewers` language.

A language `jetbrains.mps.debug.customViewers` enables one to write one's own viewers for data of certain form. During a debug session, a raw data from stack comes in special form: as proxies for values in target JVM. Such proxies are reflected in `customViewers` language with language constructs and types.

A main concept of `customViewers` language is a custom data viewer. It receives a raw java value (which comes from objects on stack) and returns a list of so-called watchables. A watchable is a pair of a value and its label (a string which categorizes a value, i.e. whether a value is a method, a field, an element, a size etc.)

The types introduced in `customViewers` language are:

- value, its descendants:
- `arrayValue`,
- `primitiveValue`,
- `objectValue`, which in turn has descendant:
- `stringValue`;
- watchable, a different type.

In the following table those types are described in detail:

Type name	Operations
<code>arrayValue</code>	<ul style="list-style-type: none">• <code>element</code> – returns value by index;• <code>allElements</code> – returns <code>list<value></code>;• <code>elementsRange</code> – returns elements from first index to second index as <code>list<value></code>;• <code>size</code> – returns size of an array as <code>int</code>.
<code>primitiveValue</code>	<code>javaValue</code> – returns an <code>Object</code> (which in fact is <code>int</code> or <code>long</code> or <code>char</code> etc); this is a java value which is reflected by this <code>primitiveValue</code> .
<code>objectValue</code>	<ul style="list-style-type: none">• <code>field</code> – gets a field's name and returns a value of that field (as "value");• <code>fields</code> – returns all fields as <code>list<value></code>;• <code>call method</code> – takes method's name and its JNI signature and returns the result of method call (as "value"); <code>call method</code> operation also receives method arguments but currently only those of primitive type (because objects can't be just written as java code executed in MPS JVM, but they should be created somehow within target JVM, and currently there's no possibility in <code>customViewers</code> language to do so; however it's planned to implement);• <code>classFQName</code> – returns a string which is an object's class' fq name;• <code>is instance of</code> – takes class fq name and returns whether object is instance of that class or not.
<code>stringValue</code>	<ul style="list-style-type: none">• all the operations of <code>objectValue</code>;• <code>javaStringValue</code> which returns a string which is equal to the string reflected by this <code>stringValue</code>.

This is the custom viewer specification for `java.util.List` class:

```
custom viewer SequentialListViewer
```

```
get value presentation:
```

```
original presentation
```

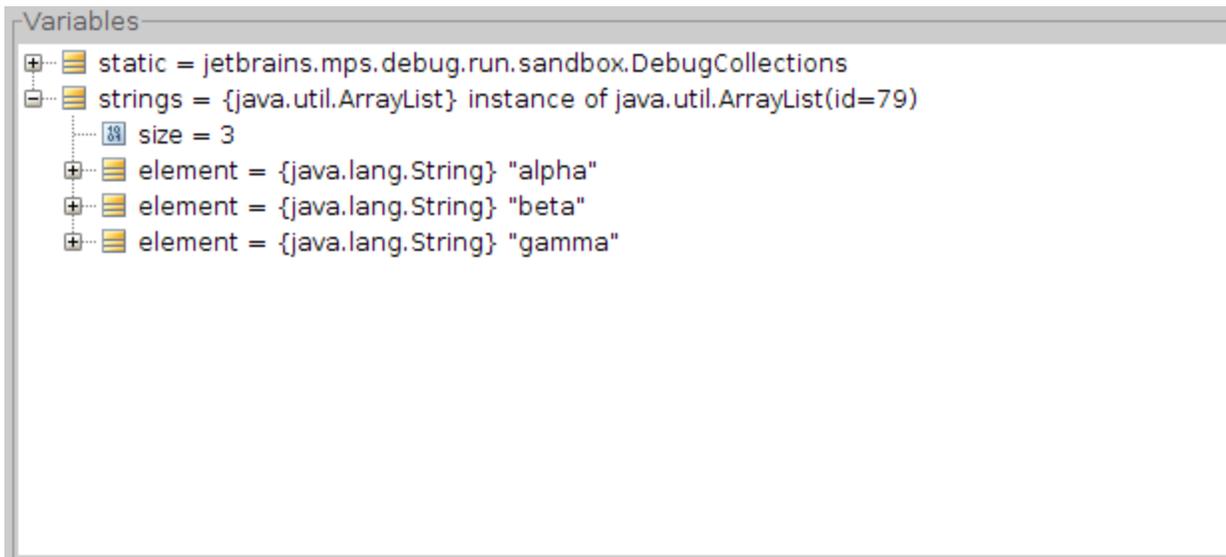
```
can wrap value:
```

```
(originalValue)->boolean {  
    if (originalValue instanceof objectValue) {  
        objectValue ov = (objectValue) originalValue;  
        return !("jetbrains.mps.internal.collections.runtime.ListSequence".equals(ov.className) ||  
            ov.is instance of ( "java.util.List" ));  
    } else {  
        return false;  
    }  
}
```

```
get custom watchables:
```

```
(originalValue)->list<watchable> {  
    objectValue objectOriginalValue = (objectValue) originalValue;  
    list<watchable> result = new arraylist<watchable>;  
  
    primitiveValue size = (primitiveValue) objectOriginalValue.call method "size" : "(  
    result.add(new watchable size ( size ));  
  
    objectValue iterator = (objectValue) objectOriginalValue.  
        call method "iterator" : "()Ljava/util/Iterator;" ( << ... >> );  
  
    while ((Boolean) ((primitiveValue) iterator.call method "hasNext" : "()Z" ( << ...  
        value value = iterator.call method "next" : "()Ljava/lang/Object;" ( << ... >> )  
        result.add(new watchable element ( value ));  
    }  
  
    return result;  
}
```

And here we see how a list is displayed in debugger view:



Creating a non-java debugger

You can create a non-java debugger using the API provided by MPS. You can see how it is done in `jetbrains.mps.samples.nanoc` language – a toy language generated into C. This language is supplied with MPS among other sample languages. The project location is `%HOME_PATH%/MPSSamples.1.5/nanoc/nanocProject/nanocProject.mpr`.

About `DebugInfoInitializer`

`DebugInfoInitializer` concept serves two purposes:

- specify which nodes require to save some additional information in `.debug` file for (like information about positions text, generated from the node, visible variables, name of the file the node was generated into etc.);
- specify how to create a breakpoint on a node.

 In MPS2.0 this is done in two different concepts.

`.debug` files

 In MPS2.0 `.debug` files were renamed to `trace.info` files.

`.debug` files contain information allowing to connect nodes in MPS with generated text. For example, if a breakpoint is hit, java debugger tells MPS the line number in source file and to get the actual node from this information MPS uses information from `.debug` files.

`.debug` files contain the following information:

- position information: name of text file and position in it where the node was generated;
- scope information: for each "scope" node (such that has some variables, associated with it and visible in the scope of the node) – names and ids of variables visible in the scope;
- unit information: for each "unit node" (such that represent some unit of a language, for example a class in java) – name of the unit the node is generated into.

How to write `DebugInfoInitializer`

 In MPS2.0 concepts `TraceableConcept`, `ScopeConcept` and `UnitConcept` of `textGen` language are used for that purpose. See [MPS 2.0 documentation](#).

To write a `DebugInfoInitializer` for your language create an instance of `DebugInfoInitializer` concept in the language's plugin.

In the following table sections of `DebugInfoInitializer` are described:

Section	Description	Example
concepts to add debug info	Concepts for which location in text is saved and for which breakpoints could be created.	<pre>5555555 create breakpoint { int a=1; } 1 return new Object() { int a=1; }</pre>
scope concepts	Concepts which have some local variables, visible in the scope.	<pre>1 int a=1; 2 int b=2; 3 int c=3; 4 int d=4; 5 int e=5; 6 int f=6; 7 int g=7; 8 int h=8; 9 int i=9; 10 int j=10; 11 int k=11; 12 int l=12; 13 int m=13; 14 int n=14; 15 int o=15; 16 int p=16; 17 int q=17; 18 int r=18; 19 int s=19; 20 int t=20; 21 int u=21; 22 int v=22; 23 int w=23; 24 int x=24; 25 int y=25; 26 int z=26; 27 int aa=27; 28 int bb=28; 29 int cc=29; 30 int dd=30; 31 int ee=31; 32 int ff=32; 33 int gg=33; 34 int hh=34; 35 int ii=35; 36 int jj=36; 37 int kk=37; 38 int ll=38; 39 int mm=39; 40 int nn=40; 41 int oo=41; 42 int pp=42; 43 int qq=43; 44 int rr=44; 45 int ss=45; 46 int tt=46; 47 int uu=47; 48 int vv=48; 49 int ww=49; 50 int xx=50; 51 int yy=51; 52 int zz=52; 53 int aaa=53; 54 int bbb=54; 55 int ccc=55; 56 int ddd=56; 57 int eee=57; 58 int fff=58; 59 int ggg=59; 60 int hhh=60; 61 int iii=61; 62 int jjj=62; 63 int kkk=63; 64 int lll=64; 65 int mmm=65; 66 int nnn=66; 67 int ooo=67; 68 int ppp=68; 69 int qqq=69; 70 int rrr=70; 71 int sss=71; 72 int ttt=72; 73 int uuu=73; 74 int vvv=74; 75 int www=75; 76 int xxx=76; 77 int yyy=77; 78 int zzz=78; 79 int aaa=79; 80 int bbb=80; 81 int ccc=81; 82 int ddd=82; 83 int eee=83; 84 int fff=84; 85 int ggg=85; 86 int hhh=86; 87 int iii=87; 88 int jjj=88; 89 int kkk=89; 90 int lll=90; 91 int mmm=91; 92 int nnn=92; 93 int ooo=93; 94 int ppp=94; 95 int qqq=95; 96 int rrr=96; 97 int sss=97; 98 int ttt=98; 99 int uuu=99; 100 int vvv=100; 101 int www=101; 102 int xxx=102; 103 int yyy=103; 104 int zzz=104; 105 int aaa=105; 106 int bbb=106; 107 int ccc=107; 108 int ddd=108; 109 int eee=109; 110 int fff=110; 111 int ggg=111; 112 int hhh=112; 113 int iii=113; 114 int jjj=114; 115 int kkk=115; 116 int lll=116; 117 int mmm=117; 118 int nnn=118; 119 int ooo=119; 120 int ppp=120; 121 int qqq=121; 122 int rrr=122; 123 int sss=123; 124 int ttt=124; 125 int uuu=125; 126 int vvv=126; 127 int www=127; 128 int xxx=128; 129 int yyy=129; 130 int zzz=130; 131 int aaa=131; 132 int bbb=132; 133 int ccc=133; 134 int ddd=134; 135 int eee=135; 136 int fff=136; 137 int ggg=137; 138 int hhh=138; 139 int iii=139; 140 int jjj=140; 141 int kkk=141; 142 int lll=142; 143 int mmm=143; 144 int nnn=144; 145 int ooo=145; 146 int ppp=146; 147 int qqq=147; 148 int rrr=148; 149 int sss=149; 150 int ttt=150; 151 int uuu=151; 152 int vvv=152; 153 int www=153; 154 int xxx=154; 155 int yyy=155; 156 int zzz=156; 157 int aaa=157; 158 int bbb=158; 159 int ccc=159; 160 int ddd=160; 161 int eee=161; 162 int fff=162; 163 int ggg=163; 164 int hhh=164; 165 int iii=165; 166 int jjj=166; 167 int kkk=167; 168 int lll=168; 169 int mmm=169; 170 int nnn=170; 171 int ooo=171; 172 int ppp=172; 173 int qqq=173; 174 int rrr=174; 175 int sss=175; 176 int ttt=176; 177 int uuu=177; 178 int vvv=178; 179 int www=179; 180 int xxx=180; 181 int yyy=181; 182 int zzz=182; 183 int aaa=183; 184 int bbb=184; 185 int ccc=185; 186 int ddd=186; 187 int eee=187; 188 int fff=188; 189 int ggg=189; 190 int hhh=190; 191 int iii=191; 192 int jjj=192; 193 int kkk=193; 194 int lll=194; 195 int mmm=195; 196 int nnn=196; 197 int ooo=197; 198 int ppp=198; 199 int qqq=199; 200 int rrr=200; 201 int sss=201; 202 int ttt=202; 203 int uuu=203; 204 int vvv=204; 205 int www=205; 206 int xxx=206; 207 int yyy=207; 208 int zzz=208; 209 int aaa=209; 210 int bbb=210; 211 int ccc=211; 212 int ddd=212; 213 int eee=213; 214 int fff=214; 215 int ggg=215; 216 int hhh=216; 217 int iii=217; 218 int jjj=218; 219 int kkk=219; 220 int lll=220; 221 int mmm=221; 222 int nnn=222; 223 int ooo=223; 224 int ppp=224; 225 int qqq=225; 226 int rrr=226; 227 int sss=227; 228 int ttt=228; 229 int uuu=229; 230 int vvv=230; 231 int www=231; 232 int xxx=232; 233 int yyy=233; 234 int zzz=234; 235 int aaa=235; 236 int bbb=236; 237 int ccc=237; 238 int ddd=238; 239 int eee=239; 240 int fff=240; 241 int ggg=241; 242 int hhh=242; 243 int iii=243; 244 int jjj=244; 245 int kkk=245; 246 int lll=246; 247 int mmm=247; 248 int nnn=248; 249 int ooo=249; 250 int ppp=250; 251 int qqq=251; 252 int rrr=252; 253 int sss=253; 254 int ttt=254; 255 int uuu=255; 256 int vvv=256; 257 int www=257; 258 int xxx=258; 259 int yyy=259; 260 int zzz=260; 261 int aaa=261; 262 int bbb=262; 263 int ccc=263; 264 int ddd=264; 265 int eee=265; 266 int fff=266; 267 int ggg=267; 268 int hhh=268; 269 int iii=269; 270 int jjj=270; 271 int kkk=271; 272 int lll=272; 273 int mmm=273; 274 int nnn=274; 275 int ooo=275; 276 int ppp=276; 277 int qqq=277; 278 int rrr=278; 279 int sss=279; 280 int ttt=280; 281 int uuu=281; 282 int vvv=282; 283 int www=283; 284 int xxx=284; 285 int yyy=285; 286 int zzz=286; 287 int aaa=287; 288 int bbb=288; 289 int ccc=289; 290 int ddd=290; 291 int eee=291; 292 int fff=292; 293 int ggg=293; 294 int hhh=294; 295 int iii=295; 296 int jjj=296; 297 int kkk=297; 298 int lll=298; 299 int mmm=299; 300 int nnn=300; 301 int ooo=301; 302 int ppp=302; 303 int qqq=303; 304 int rrr=304; 305 int sss=305; 306 int ttt=306; 307 int uuu=307; 308 int vvv=308; 309 int www=309; 310 int xxx=310; 311 int yyy=311; 312 int zzz=312; 313 int aaa=313; 314 int bbb=314; 315 int ccc=315; 316 int ddd=316; 317 int eee=317; 318 int fff=318; 319 int ggg=319; 320 int hhh=320; 321 int iii=321; 322 int jjj=322; 323 int kkk=323; 324 int lll=324; 325 int mmm=325; 326 int nnn=326; 327 int ooo=327; 328 int ppp=328; 329 int qqq=329; 330 int rrr=330; 331 int sss=331; 332 int ttt=332; 333 int uuu=333; 334 int vvv=334; 335 int www=335; 336 int xxx=336; 337 int yyy=337; 338 int zzz=338; 339 int aaa=339; 340 int bbb=340; 341 int ccc=341; 342 int ddd=342; 343 int eee=343; 344 int fff=344; 345 int ggg=345; 346 int hhh=346; 347 int iii=347; 348 int jjj=348; 349 int kkk=349; 350 int lll=350; 351 int mmm=351; 352 int nnn=352; 353 int ooo=353; 354 int ppp=354; 355 int qqq=355; 356 int rrr=356; 357 int sss=357; 358 int ttt=358; 359 int uuu=359; 360 int vvv=360; 361 int www=361; 362 int xxx=362; 363 int yyy=363; 364 int zzz=364; 365 int aaa=365; 366 int bbb=366; 367 int ccc=367; 368 int ddd=368; 369 int eee=369; 370 int fff=370; 371 int ggg=371; 372 int hhh=372; 373 int iii=373; 374 int jjj=374; 375 int kkk=375; 376 int lll=376; 377 int mmm=377; 378 int nnn=378; 379 int ooo=379; 380 int ppp=380; 381 int qqq=381; 382 int rrr=382; 383 int sss=383; 384 int ttt=384; 385 int uuu=385; 386 int vvv=386; 387 int www=387; 388 int xxx=388; 389 int yyy=389; 390 int zzz=390; 391 int aaa=391; 392 int bbb=392; 393 int ccc=393; 394 int ddd=394; 395 int eee=395; 396 int fff=396; 397 int ggg=397; 398 int hhh=398; 399 int iii=399; 400 int jjj=400; 401 int kkk=401; 402 int lll=402; 403 int mmm=403; 404 int nnn=404; 405 int ooo=405; 406 int ppp=406; 407 int qqq=407; 408 int rrr=408; 409 int sss=409; 410 int ttt=410; 411 int uuu=411; 412 int vvv=412; 413 int www=413; 414 int xxx=414; 415 int yyy=415; 416 int zzz=416; 417 int aaa=417; 418 int bbb=418; 419 int ccc=419; 420 int ddd=420; 421 int eee=421; 422 int fff=422; 423 int ggg=423; 424 int hhh=424; 425 int iii=425; 426 int jjj=426; 427 int kkk=427; 428 int lll=428; 429 int mmm=429; 430 int nnn=430; 431 int ooo=431; 432 int ppp=432; 433 int qqq=433; 434 int rrr=434; 435 int sss=435; 436 int ttt=436; 437 int uuu=437; 438 int vvv=438; 439 int www=439; 440 int xxx=440; 441 int yyy=441; 442 int zzz=442; 443 int aaa=443; 444 int bbb=444; 445 int ccc=445; 446 int ddd=446; 447 int eee=447; 448 int fff=448; 449 int ggg=449; 450 int hhh=450; 451 int iii=451; 452 int jjj=452; 453 int kkk=453; 454 int lll=454; 455 int mmm=455; 456 int nnn=456; 457 int ooo=457; 458 int ppp=458; 459 int qqq=459; 460 int rrr=460; 461 int sss=461; 462 int ttt=462; 463 int uuu=463; 464 int vvv=464; 465 int www=465; 466 int xxx=466; 467 int yyy=467; 468 int zzz=468; 469 int aaa=469; 470 int bbb=470; 471 int ccc=471; 472 int ddd=472; 473 int eee=473; 474 int fff=474; 475 int ggg=475; 476 int hhh=476; 477 int iii=477; 478 int jjj=478; 479 int kkk=479; 480 int lll=480; 481 int mmm=481; 482 int nnn=482; 483 int ooo=483; 484 int ppp=484; 485 int qqq=485; 486 int rrr=486; 487 int sss=487; 488 int ttt=488; 489 int uuu=489; 490 int vvv=490; 491 int www=491; 492 int xxx=492; 493 int yyy=493; 494 int zzz=494; 495 int aaa=495; 496 int bbb=496; 497 int ccc=497; 498 int ddd=498; 499 int eee=499; 500 int fff=500; 501 int ggg=501; 502 int hhh=502; 503 int iii=503; 504 int jjj=504; 505 int kkk=505; 506 int lll=506; 507 int mmm=507; 508 int nnn=508; 509 int ooo=509; 510 int ppp=510; 511 int qqq=511; 512 int rrr=512; 513 int sss=513; 514 int ttt=514; 515 int uuu=515; 516 int vvv=516; 517 int www=517; 518 int xxx=518; 519 int yyy=519; 520 int zzz=520; 521 int aaa=521; 522 int bbb=522; 523 int ccc=523; 524 int ddd=524; 525 int eee=525; 526 int fff=526; 527 int ggg=527; 528 int hhh=528; 529 int iii=529; 530 int jjj=530; 531 int kkk=531; 532 int lll=532; 533 int mmm=533; 534 int nnn=534; 535 int ooo=535; 536 int ppp=536; 537 int qqq=537; 538 int rrr=538; 539 int sss=539; 540 int ttt=540; 541 int uuu=541; 542 int vvv=542; 543 int www=543; 544 int xxx=544; 545 int yyy=545; 546 int zzz=546; 547 int aaa=547; 548 int bbb=548; 549 int ccc=549; 550 int ddd=550; 551 int eee=551; 552 int fff=552; 553 int ggg=553; 554 int hhh=554; 555 int iii=555; 556 int jjj=556; 557 int kkk=557; 558 int lll=558; 559 int mmm=559; 560 int nnn=560; 561 int ooo=561; 562 int ppp=562; 563 int qqq=563; 564 int rrr=564; 565 int sss=565; 566 int ttt=566; 567 int uuu=567; 568 int vvv=568; 569 int www=569; 570 int xxx=570; 571 int yyy=571; 572 int zzz=572; 573 int aaa=573; 574 int bbb=574; 575 int ccc=575; 576 int ddd=576; 577 int eee=577; 578 int fff=578; 579 int ggg=579; 580 int hhh=580; 581 int iii=581; 582 int jjj=582; 583 int kkk=583; 584 int lll=584; 585 int mmm=585; 586 int nnn=586; 587 int ooo=587; 588 int ppp=588; 589 int qqq=589; 590 int rrr=590; 591 int sss=591; 592 int ttt=592; 593 int uuu=593; 594 int vvv=594; 595 int www=595; 596 int xxx=596; 597 int yyy=597; 598 int zzz=598; 599 int aaa=599; 600 int bbb=600; 601 int ccc=601; 602 int ddd=602; 603 int eee=603; 604 int fff=604; 605 int ggg=605; 606 int hhh=606; 607 int iii=607; 608 int jjj=608; 609 int kkk=609; 610 int lll=610; 611 int mmm=611; 612 int nnn=612; 613 int ooo=613; 614 int ppp=614; 615 int qqq=615; 616 int rrr=616; 617 int sss=617; 618 int ttt=618; 619 int uuu=619; 620 int vvv=620; 621 int www=621; 622 int xxx=622; 623 int yyy=623; 624 int zzz=624; 625 int aaa=625; 626 int bbb=626; 627 int ccc=627; 628 int ddd=628; 629 int eee=629; 630 int fff=630; 631 int ggg=631; 632 int hhh=632; 633 int iii=633; 634 int jjj=634; 635 int kkk=635; 636 int lll=636; 637 int mmm=637; 638 int nnn=638; 639 int ooo=639; 640 int ppp=640; 641 int qqq=641; 642 int rrr=642; 643 int sss=643; 644 int ttt=644; 645 int uuu=645; 646 int vvv=646; 647 int www=647; 648 int xxx=648; 649 int yyy=649; 650 int zzz=650; 651 int aaa=651; 652 int bbb=652; 653 int ccc=653; 654 int ddd=654; 655 int eee=655; 656 int fff=656; 657 int ggg=657; 658 int hhh=658; 659 int iii=659; 660 int jjj=660; 661 int kkk=661; 662 int lll=662; 663 int mmm=663; 664 int nnn=664; 665 int ooo=665; 666 int ppp=666; 667 int qqq=667; 668 int rrr=668; 669 int sss=669; 670 int ttt=670; 671 int uuu=671; 672 int vvv=672; 673 int www=673; 674 int xxx=674; 675 int yyy=675; 676 int zzz=676; 677 int aaa=677; 678 int bbb=678; 679 int ccc=679; 680 int ddd=680; 681 int eee=681; 682 int fff=682; 683 int ggg=683; 684 int hhh=684; 685 int iii=685; 686 int jjj=686; 687 int kkk=687; 688 int lll=688; 689 int mmm=689; 690 int nnn=690; 691 int ooo=691; 692 int ppp=692; 693 int qqq=693; 694 int rrr=694; 695 int sss=695; 696 int ttt=696; 697 int uuu=697; 698 int vvv=698; 699 int www=699; 700 int xxx=700; 701 int yyy=701; 702 int zzz=702; 703 int aaa=703; 704 int bbb=704; 705 int ccc=705; 706 int ddd=706; 707 int eee=707; 708 int fff=708; 709 int ggg=709; 710 int hhh=710; 711 int iii=711; 712 int jjj=712; 713 int kkk=713; 714 int lll=714; 715 int mmm=715; 716 int nnn=716; 717 int ooo=717; 718 int ppp=718; 719 int qqq=719; 720 int rrr=720; 721 int sss=721; 722 int ttt=722; 723 int uuu=723; 724 int vvv=724; 725 int www=725; 726 int xxx=726; 727 int yyy=727; 728 int zzz=728; 729 int aaa=729; 730 int bbb=730; 731 int ccc=731; 732 int ddd=732; 733 int eee=733; 734 int fff=734; 735 int ggg=735; 736 int hhh=736; 737 int iii=737; 738 int jjj=738; 739 int kkk=739; 740 int lll=740; 741 int mmm=741; 742 int nnn=742; 743 int ooo=743; 744 int ppp=744; 745 int qqq=745; 746 int rrr=746; 747 int sss=747; 748 int ttt=748; 749 int uuu=749; 750 int vvv=750; 751 int www=751; 752 int xxx=752; 753 int yyy=753; 754 int zzz=754; 755 int aaa=755; 756 int bbb=756; 757 int ccc=757; 758 int ddd=758; 759 int eee=759; 760 int fff=760; 761 int ggg=761; 762 int hhh=762; 763 int iii=763; 764 int jjj=764; 765 int kkk=765; 766 int lll=766; 767 int mmm=767; 768 int nnn=768; 769 int ooo=769; 770 int ppp=770; 771 int qqq=771; 772 int rrr=772; 773 int sss=773; 774 int ttt=774; 775 int uuu=775; 776 int vvv=776; 777 int www=777; 778 int xxx=778; 779 int yyy=779; 780 int zzz=780; 781 int aaa=781; 782 int bbb=782; 783 int ccc=783; 784 int ddd=784; 785 int eee=785; 786 int fff=786; 787 int ggg=787; 788 int hhh=788; 789 int iii=789; 790 int jjj=790; 791 int kkk=791; 792 int lll=792; 793 int mmm=793; 794 int nnn=794; 795 int ooo=795; 796 int ppp=796; 797 int qqq=797; 798 int rrr=798; 799 int sss=799; 800 int ttt=800; 801 int uuu=801; 802 int vvv=802; 803 int www=803; 804 int xxx=804; 805 int yyy=805; 806 int zzz=806; 807 int aaa=807; 808 int bbb=808; 809 int ccc=809; 810 int ddd=810; 811 int eee=811; 812 int fff=812; 813 int ggg=813; 814 int hhh=814; 815 int iii=815; 816 int jjj=816; 817 int kkk=817; 818 int lll=818; 819 int mmm=819; 820 int nnn=820; 821 int ooo=821; 822 int ppp=822; 823 int qqq=823; 824 int rrr=824; 825 int sss=825; 826 int ttt=826; 827 int uuu=827; 828 int vvv=828; 829 int www=829; 830 int xxx=830; 831 int yyy=831; 832 int zzz=832; 833 int aaa=833; 834 int bbb=834; 835 int ccc=835; 836 int ddd=836; 837 int eee=837; 838 int fff=838; 839 int ggg=839; 840 int hhh=840; 841 int iii=841; 842 int jjj=842; 843 int kkk=843; 844 int lll=844; 845 int mmm=845; 846 int nnn=846; 847 int ooo=847; 848 int ppp=848; 849 int qqq=849; 850 int rrr=850; 851 int sss=851; 852 int ttt=852; 853 int uuu=853; 854 int vvv=854; 855 int www=855; 856 int xxx=856; 857 int yyy=857; 858 int zzz=858; 859 int aaa=859; 860 int bbb=860; 861 int ccc=861; 862 int ddd=862; 863 int eee=863; 864 int fff=864; 865 int ggg=865; 866 int hhh=866; 867 int iii=867; 868 int jjj=868; 869 int kkk=869; 870 int lll=870; 871 int mmm=871; 872 int nnn=872; 873 int ooo=873; 874 int ppp=874; 875 int qqq=875; 876 int rrr=876; 877 int sss=877; 878 int ttt=878; 879 int uuu=879; 880 int vvv=880; 881 int www=881; 882 int xxx=882; 883 int yyy=883; 884 int zzz=884; 885 int aaa=885; 886 int bbb=886; 887 int ccc=887; 888 int ddd=888; 889 int eee=889; 890 int fff=890; 891 int ggg=891; 892 int hhh=892; 893 int iii=893; 894 int jjj=894; 895 int kkk=895; 896 int lll=896; 897 int mmm=897; 898 int nnn=898; 899 int ooo=899; 900 int ppp=900; 901 int qqq=901; 902 int rrr=902; 903 int sss=903; 904 int ttt=904; 905 int uuu=905; 906 int vvv=906; 907 int www=907; 908 int xxx=908; 909 int yyy=909; 910 int zzz=910; 911 int aaa=911; 912 int bbb=912; 913 int ccc=913; 914 int ddd=914; 915 int eee=915; 916 int fff=916; 917 int ggg=917; 918 int hhh=918; 919 int iii=919; 920 int jjj=920; 921 int kkk=921; 922 int lll=922; 923 int mmm=923; 924 int nnn=924; 925 int ooo=925; 926 int ppp=926; 927 int qqq=927; 928 int rrr=928; 929 int sss=929; 930 int ttt=930; 931 int uuu=931; 932 int vvv=932; 933 int www=933; 934 int xxx=934; 935 int yyy=935; 936 int zzz=936; 937 int aaa=937; 938 int bbb=938; 939 int ccc=939; 940 int ddd=940; 941 int eee=941; 942 int fff=942; 943 int ggg=943; 944 int hhh=944; 945 int iii=945; 946 int jjj=946; 947 int kkk=947; 948 int lll=948; 949 int mmm=949; 950 int nnn=950; 951 int ooo=951; 952 int ppp=952; 953 int qqq=953; 954 int rrr=954; 955 int sss=955; 956 int ttt=956; 957 int uuu=957; 958 int vvv=958; 959 int www=959; 960 int xxx=960; 961 int yyy=961; 962 int zzz=962; 963 int aaa=963; 964 int bbb=964; 965 int ccc=965; 966 int ddd=966; 967 int eee=967; 968 int fff=968; 969 int ggg=969; 970 int hhh=970; 971 int iii=971; 972 int jjj=972; 973 int kkk=973; 974 int lll=974; 975 int mmm=975; 976 int nnn=976; 977 int ooo=977; 978 int ppp=978; 979 int qqq=979; 980 int rrr=980; 981 int sss=981; 982 int ttt=982; 983 int uuu=983; 984 int vvv=984; 985 int www=985; 986 int xxx=986; 987 int yyy=987; 988 int zzz=988; 989 int aaa=989; 990 int bbb=990; 991 int ccc=991; 992 int ddd=992; 993 int eee=993; 994 int fff=994; 995 int ggg=995; 996 int hhh=996; 997 int iii=997; 998 int jjj=998; 999 int kkk=999; 1000 int lll=1000; 1001 int mmm=1001; 1002 int nnn=1002; 1003 int ooo=1003; 1004 int ppp=1004; 1005 int qqq=1005; 1006 int rrr=1006; 1007 int sss=1007; 1008 int ttt=1008; 1009 int uuu=1009; 1010 int vvv=1010; 1011 int www=1011; 1012 int xxx=1012; 1013 int yyy=1013; 1014 int zzz=1014; 1015 int aaa=1015; 1016 int bbb=1016; 1017 int ccc=1017; 1018 int ddd=1018; 1019 int eee=1019; 1020 int fff=1020; 1021 int ggg=1021; 1022 int hhh=1022; 1023 int iii=1023; 1024 int jjj=1024; 1025 int kkk=1025; 1026 int lll=1026; 1027 int mmm=1027; 1028 int nnn=1028; 1029 int ooo=1029; 1030 int ppp=1030; 1031 int qqq=1031; 1032 int rrr=1032; 1033 int sss=1033; 1034 int ttt=1034; 1035 int uuu=1035; 1036 int vvv=1036; 1037 int www=1037; 1038 int xxx=1038; 1039 int yyy=1039; 1040 int zzz=1040; 1041 int aaa=1041; 1042 int bbb=1042; 1043 int ccc=1043; 1044 int ddd=1044; 104</pre>