Documentum Methods Troubleshooting Guide.

June 2003.

Contents

Introduction

The purpose of this troubleshooting guide is to assist Documentum users determine the failure of methods when using eContent Server 5.x or 4.2.x Once a few of these tests are made, it will be easier to determine if a specific method is failing or if all of your methods running on the server are failing.

Finally this guide is intended to bring together previous published technical support notes into one publication into an organized presentation.

Methods

Methods are executable scripts or programs that are represented by method objects in the Docbase. The script or program can be a Docbasic script, a Java method, or a program written in another programming language such as C++.

The associated method object has attributes that identify the executable and define command line arguments, and the execution parameters. Methods are executed by issuing a DO_METHOD administrative method or using a job. Using a DO_METHOD allows you to execute the method on demand. Using a job allows you to schedule the method for regular, automatic execution.

The executable invoked by the method can be stored in an external file or as content of the method object or it can be installed on the application server. How it is stored depends on the program:

- ✓ If the program does not require an interpretive language to run it or if it has a C shell wrapper, store the program in an external file.
- ✓ If the program is a script that requires an interpretive language to run it, store the program as the content of the associated method object.
- ✓ If the program is a Java method and you want to execute it on the application server, install it on the application server.

Execution Agents

The execution agents are the three server processes that are capable of executing methods. There are three execution agents:

- ✓ Method Server
- ✓ Application Server
- ✓ Content server

Method Server

The method server is a separate process that is installed with Content Server and resides on the same host. After it is enabled and started, it runs continuously. If you stop Content Server, the method server is also stopped. If the method server stops, Content Server restarts it automatically.

Application Server

Documentum provides the Apache Tomcat application server as an optional component of the Content Server installation. In addition to Tomcat, Documentum provides a servlet to execute methods called by DO_METHOD. The servlet is referenced in the server config's app_server_name attribute as "do_method".

The application server resides on the same host as Content Server. (However, it is not started, or stopped, automatically when Content Server starts and stops. It must be manually started and stopped.)

Content Server

Documentum Content Server is the default execution agent for methods if you do not set the method's use_method_server attribute to TRUE to direct the execution to the method server or application server.

There are some parameters that will help you to enable those methods in the server.ini file such

method_server_enabled
The method_server_enabled key is a Boolean key that controls whether the method server or an application server may be used to execute dm_method objects. The default value is F (FALSE), meaning that dm_method objects are executed through Content Server.

Note: Enabling the method server alone does not cause dm_method objects to be executed by the method server or application server. The method objects must also be configured correctly.

method_server_threads
The method_server_threads key defines the maximum number of method server worker processes are available to execute method objects. The default (and minimum) value is 5. The maximum value for this key is the value set in the concurrent_sessions attribute of the server config object.

Execution Method

The execution agents can execute:

- ✓ Use the method server to execute Docbasic scripts.
- ✓ Use Content Server to execute Docbasic scripts or programs in any language.
- ✓ Use an application server to execute Java methods that are installed on the application server.

To execute Java or DFC calls using the method server, the Java or DFC calls must be called in a Docbasic program.

You cannot use the application server to execute Docbasic scripts or any program other than Java methods installed on the application server.

Content Server is the default execution agent for methods.

Tracing Options for Methods

You can trace the DO_METHOD administrative method, the program execution, or both.

To trace the DO_METHOD method, set the trace_launch attribute of the method object or the TRACE_LAUNCH argument on the DO_METHOD command line. If both are set, the setting on the command line overrides the attribute setting. The trace information includes the command executed to invoke the method and messages indicating the success or failure of the invocation. The trace information generated by TRACE_LAUNCH is stored in the Docbase log file. Tracing information generated by Content Server is stored in the server log file.

The log file is stored in $DOCUMENTUM/dba/log/Docbase.log

To trace method execution in the method server or Content Server, set the trace_method_server server flag.

The log file is stored in $DOCUMENTUM/dba/log/<*docbase_id*>/MethodServer/MethodServer.log

To trace method execution on the application server, set the trace parameter in the web.xml file to true. For example:

```
<init-param>
        <param-name>trace</param-name>
        <param-value>t</param-value>
</init-param>
```

The web.xml file is found in the WEB_INF folder that is created when you install and configure Apache Tomcat, the application server.

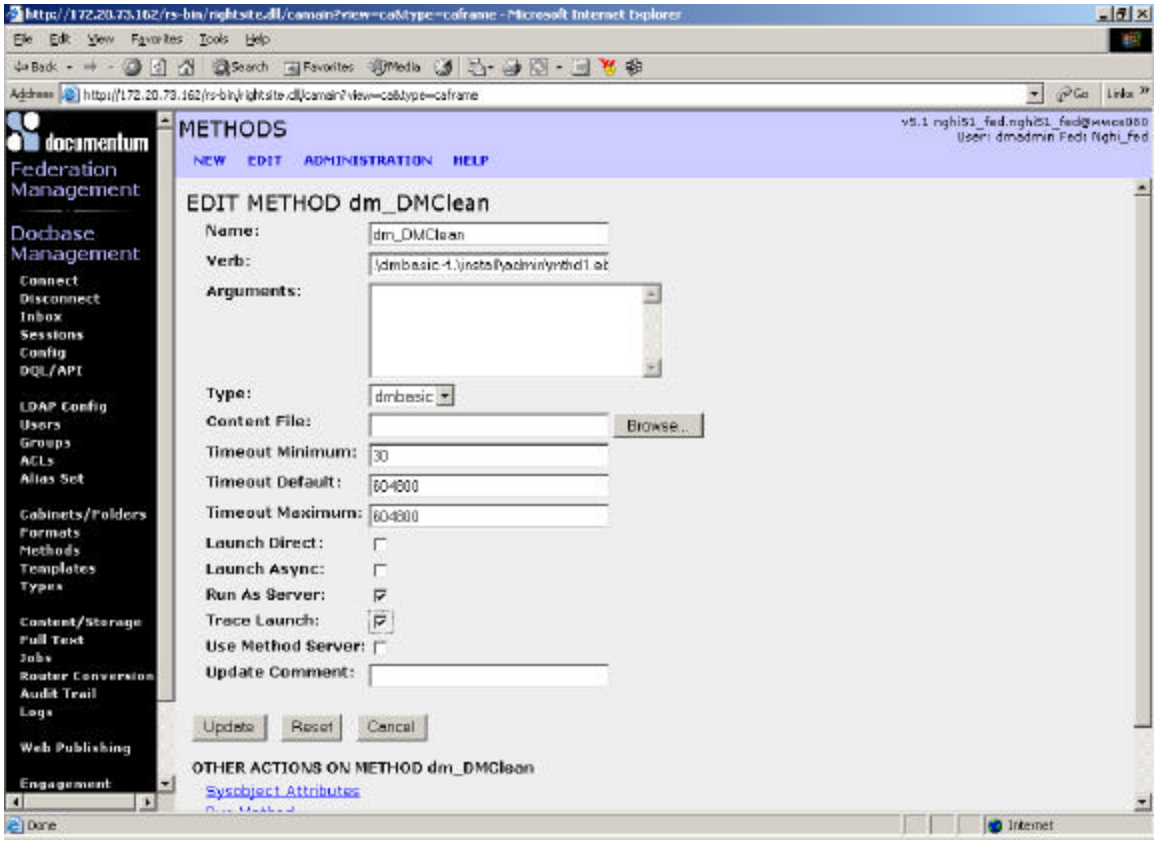
Example of the output:

```
TRACE LAUNCH: .\dmbasic -f..\install\admin\mthd1.ebs -eDMClean  -h 984 -- -docbase_name
MiguelNT414o.MiguelNT414o -user_name zarate -job_id 08001a98800001db -method_trace_level 10

dmExec::Launch(D:\DOCUMENTUM\product\4.2\bin\.\dmbasic -f..\install\admin\mthd1.ebs -eDMClean  -h
984 -- -docbase_name MiguelNT414o.MiguelNT414o -user_name zarate -job_id 08001a98800001db -
method_trace_level 10 )

Working Directory(D:\DOCUMENTUM\product\4.2\bin)
```

Troubleshooting

Here are the steps to troubleshoot methods:

1. The first thing that we need to make sure is that the method is launching and we can find out by setting the trace_launch to True in the method.



2. Dump the method attributes to find out f the method verb or the timeout settings are correct:

Go to iapi:

```
API> retrieve,c,dm_method where object_name='dm_DMClean'
...
10001a98800001dc
API> dump,c,10001a98800001dc
...
USER ATTRIBUTES

  object_name             : dm_DMClean
  title                   :
  subject                 :
  authors                 []: <none>
  keywords                []: <none>
  resolution_label        :
  owner_name              : zarate
```

```
owner_permit            : 7
group_name              : docu
group_permit            : 1
world_permit            : 1
log_entry               : test
acl_domain              : zarate
acl_name                : dm_acl_superusers
language_code           :
method_verb             : .\dmbasic -f..\install\admin\mthd1.ebs
                  -eDMClean
method_args             []: <none>
timeout_min             : 30
timeout_max             : 604800
timeout_default         : 604800
launch_direct           : F
launch_async            : F
trace_launch            : T
run_as_server           : T
use_method_content      : F
method_type             : dmbasic
```

SYSTEM ATTRIBUTES

```
r_object_id             : 10001a98800001dc
r_object_type           : dm_method
r_creation_date         : 8/24/2000 10:27:14 AM
r_modify_date           : 6/24/2003 03:48:21 PM
r_modifier              : zarate
r_access_date           : nulldate
r_composite_id          []: <none>
r_composite_label       []: <none>
r_component_label       []: <none>
r_order_no              []: <none>
r_link_cnt              : 0
r_link_high_cnt         : 0
r_assembled_from_id     : 0000000000000000
r_frzn_assembly_cnt     : 0
r_has_frzn_assembly     : F
r_is_virtual_doc        : 0
r_page_cnt              : 0
r_content_size          : 0
r_lock_owner            :
r_lock_date             : nulldate
r_lock_machine          :
r_version_label         [0]: 1.0
                  [1]: CURRENT
r_immutable_flag        : F
r_frozen_flag           : F
r_has_events            : F
r_creator_name          : zarate
r_is_public             : F
r_policy_id             : 0000000000000000
r_resume_state          : 0
r_current_state         : 0
r_alias_set_id          : 0000000000000000
```

APPLICATION ATTRIBUTES

```
a_application_type      :
a_status                :
a_is_hidden             : F
a_retention_date        : nulldate
```

```
a_archive              : F
a_compound_architecture    :
a_link_resolved          : F
a_content_type           :
a_full_text            : F
a_storage_type           :
a_special_app            : 1.0.0
a_effective_date       []: <none>
a_expiration_date      []: <none>
a_publish_formats      []: <none>
a_effective_label      []: <none>
a_effective_flag       []: <none>
a_category             :

INTERNAL ATTRIBUTES

i_is_deleted           : F
i_reference_cnt          : 1
i_has_folder           : T
i_folder_id          [0]: 0b001a9880000106
i_contents_id            : 0000000000000000
i_cabinet_id           : 0c001a9880000104
i_antecedent_id          : 0000000000000000
i_chronicle_id         : 10001a98800001dc
i_latest_flag          : T
i_branch_cnt           : 0
i_direct_dsc           : F
i_is_reference           : F
i_is_replica           : F
i_vstamp               : 20

API>
```

Make sure that the following is correct:

- ✓ Method verb -- Command-line name of the procedure or the name of the interpretive language that will execute the program file. Use “dmbasic” if the program is written in Docbasic. Use “dmawk32” if the program is written in dmawk. Use the namespace in which to launch the method if the method is executing on the application server. For example, com.foo.<package_name>.<class> or fooPayMethod

For example in 4.2.x the value for dmclean should be: .\dmbasic -f..\install\admin\mthd1.ebs -eDMClean but in 5.1 it’s .\dmclean.exe

- ✓ Timeout attributes
  - o timeout_min – The minimum time-out that you can specify on the command line for this procedure. The default is 30 seconds.
  - o timeout_max -- The maximum time-out that you can specify on the command line for this procedure. The default is 300 seconds.
  - o timeout_default -- The default time-out for this procedure. Used if no other time-out is specified on the command line. The default is 60 seconds.

In 4.2.x docbases the values are:

```
timeout_min            : 30
timeout_max            : 604800
timeout_default        : 604800
```

In 5.1 docbases the values are:

```
timeout_min            : 30
timeout_max            : 86400
```

timeout_default : 86400

Those values are set when the docbase is configured and you can change the values if needed.

- ✓ Trace launch -- Controls whether internal trace messages generated by the executing program are logged. If set to TRUE, the messages are put in the session log. If set to FALSE, the messages are not logged. This is FALSE by default.
  Notes: If trace_launch is TRUE and SAVE_RESULT is set to TRUE on the command line, the trace output is put into the result document. If the method is executing on the application server, the trace records information about the method launch. Internal trace messages are stored in a result document if SAVE_RESULTS is set to TRUE.

- ✓ Use Method Content – Indicates whether the program that you want to run is stored as content in the method object or as an external file. Set this to TRUE if you have specified an interpretive language for method_verb and the program you want to run is stored as the method's content. Set this to FALSE if method_verb contains a full file path to a program or script. The value is ignored if the method is executing on an application server.

3. Then you can run the methods either from the command prompt or from the iapi:

From iapi:

```
API> apply,c,NULL,DO_METHOD,METHOD,S,dmclean,SAVE_RESULTS,B,T
...
q0
API> next,c,q0
...
Ok
API> dump,c,q0
...
USER ATTRIBUTES

  result                  : 09001a988005716a
  result_doc_id           : 09001a988005716a
  process_id              : 3424
  launch_failed           : F
  method_return_val       : 0
  os_system_error         : No Error
  timed_out               : F
  time_out_length         : 86400

SYSTEM ATTRIBUTES


APPLICATION ATTRIBUTES


INTERNAL ATTRIBUTES
```

From the command prompt:

- Get the job_id from idql and method verb:

job_id:

```
DQL> select r_object_id from dm_job where object_name='dm_DMClean';

r_object_id
================

08001a98800001db
(1 rows affected)
DQL>
```

method_verb:

```
DQL> select method_verb from dm_method where object_name='dm_DMClean';

method_verb
========================================================================

.\dmbasic -f..\install\admin\mthd1.ebs -eDMClean
(1 rows affected)
DQL>
```

Then from the command prompt:

- Goto the following directory:

/$DM_HOME/bin/

<method verb> -- -docbase_name <docbase name> -user_name <superuser name> -job_id <r_object_id of the job> trace_level 10

Example:

.\dmbasic -f..\install\admin\mthd1.ebs -eDMClean -- -docbase_name MiguelNT414o -user_name dmadmin -job_id 08001a98800001db trace_level 10

Debugging Methods in dmbasic

In order to debug methods in dmbasic you can set the following code to find out what is the result of some variable or result that you identify that has the problem:

```
Open "<file name>.txt" For Output As #99       ‘Create new file with the output
‘ Header

  print #99, "Start of output"
  print #99, "-------------------------------------------------------------"
  print #99, " "
  print #99, "Start of log:" ;dmNow(date$ & " " & time$)
  print #99, "-------------"
  Print #99, "    "


  Print #99, "ResultFile value """ & <variable to debug to see the result> & """-"
  print #99, "-------------"
```

Example:

```
  Print #99, "cmd$ value """ & cmd$ & """-"
  print #99, "-------------"

close #99                                          ‘Close output file
```

Example:

Let’s say that the dm_dmclean job is failing and you are getting the following error in the ob report:

```
----------------------------------------------------Job Report----------------------------------------------------

Connected To MiguelNT414o.MiguelNT414o
There was a problem, and the program aborted...
DMClean Tool was aborted at 6/25/2003 10:05:15.  View the job's report for details.
Calling SetJobStatus function...
[DM_SESSION_I_SESSION_START]info:  "Session 01001a9880071d17 started for user zarate."



--- Start D:\DOCUMENTUM\dba\log\00001a98\sysadmin\DMCleanDoc.txt report output ----
DMClean Report For DocBase MiguelNT414o As Of 6/25/2003 10:05:15

-custom_predicate is a bad argument.
You have one or more bad method argument(s), aborting...
Report End  6/25/2003 10:05:15
--- End D:\DOCUMENTUM\dba\log\00001a98\sysadmin\DMCleanDoc.txt report output ---

----------------------------------------------------Job Report----------------------------------------------------
```

Basically several Documentum jobs checks the arguments against the toolset.ini file located in:

$DM_HOME/install/admin

And some times the arguments apparently match each other, but what happens if there is a space in the argument or is set wrong or simply is not a valid argument?

That can cause the error above; in order to debug that and find out what is the problem with the arguments you can set the following code in the method:

In the mthd1.ebs file we have the following subroutine to check the arguments:

```
If CheckForBadArgs(JobID, tempName$) = 0 Then
  theMsg$ = "You have one or more bad method argument(s), aborting..."
  Print #2, theMsg$
  ret% = SetJobStatus(JobID, theMsg$)
  ran_ok = "false"
  Call CloseOut(start_date,ran_ok,JobName,JobID,ResultDoc,ResultPath, MethodTraceLevel)
  ret% = dmAPIExec("disconnect,c")
  Exit Sub
End If


Function CheckForBadArgs(JobID As String, theJob as string) As Integer
 Dim NumItems As String
 Dim TheString As String
 Dim numArgs As Integer
 Dim iniPath As String, ps as string
 Dim validArgs(100) as string
 Dim Fnum As Integer

 CheckForBadArgs = 1

 ' build array of valid arguments by reading toolset.ini

 ps = Basic.PathSeparator$
 iniPath = CurDir & ps & ".." & ps & "install" & ps & "admin" & ps & "toolset.ini"
 Fnum = FreeFile
 Open iniPath For Input As #Fnum
 Do While Not EOF(Fnum)
  Line Input Fnum, CurLine$
  If Item$(CurLine$,1,1," ") = theJob Then
   Line Input Fnum, CurLine$ ' skip description line
   Line Input Fnum, CurLine$ ' find out how many arguments there are
   numArgs = Cint(Item$(CurLine$, 5, 5, "|"))
   For I = 1 To numArgs
       Line Input Fnum, CurLine$
       validArgs(I) = Item$(CurLine$, 1, 1, " ")
   Next I
   Exit Do
  End If
 Loop
 Close #Fnum
```

```
    NumItems = dmAPIGet("values,s0," & JobId & ",method_arguments")
    For I = 0 To Val(NumItems) - 1
       TheString = dmAPIGet("get,s0," & JobId & ",method_arguments[" & str$(I) & "]")
       tempBadArgs% = 0
      For J = 1 To numArgs
        If validArgs(J) = Item$(TheString, 1, 1, " ") Then
          tempBadArgs% = 1
          Exit For
        End If
      Next J
      If tempBadArgs% = 0 Then
        CheckForBadArgs = 0
        Print #2, TheString & " is a bad argument."
      End If
    Next I
End Function
```

If you add the code to debug the code should look like this:

```
  If CheckForBadArgs(JobID, tempName$) = 0 Then
    theMsg$ = "You have one or more bad method argument(s), aborting..."
    Print #2, theMsg$
    ret% = SetJobStatus(JobID, theMsg$)
    ran_ok = "false"
    Call CloseOut(start_date,ran_ok,JobName,JobID,ResultDoc,ResultPath, MethodTraceLevel)
    ret% = dmAPIExec("disconnect,c")
    Exit Sub
  End If


  Function CheckForBadArgs(JobID As String, theJob as string) As Integer
    Dim NumItems As String
    Dim TheString As String
    Dim numArgs As Integer
    Dim iniPath As String, ps as string
    Dim validArgs(100) as string
    Dim Fnum As Integer

    CheckForBadArgs = 1

    ' build array of valid arguments by reading toolset.ini

    ps = Basic.PathSeparator$
    iniPath = CurDir & ps & ".." & ps & "install" & ps & "admin" & ps & "toolset.ini"
    Fnum = FreeFile
    Open iniPath For Input As #Fnum
    Do While Not EOF(Fnum)
      Line Input Fnum, CurLine$
      If Item$(CurLine$,1,1," ") = theJob Then
        Line Input Fnum, CurLine$ ' skip description line
        Line Input Fnum, CurLine$ ' find out how many arguments there are
        numArgs = Cint(Item$(CurLine$, 5, 5, "|"))
        For I = 1 To numArgs
            Line Input Fnum, CurLine$
            validArgs(I) = Item$(CurLine$, 1, 1, " ")
        Next I
        Exit Do
      End If
    Loop
    Close #Fnum
```

```
‘MODIFICATION ADDING CODE TO DEBUG
‘CREATE A NEW FILE TO SHOW THE OUTPUT AND ADD HEADER
      Open "clean.txt" For Output As #99
      print #99, "Start of output"
      print #99, "-----------------------------------------------------------"
      print #99, " "
      print #99, "Start of log:" ;dmNow(date$ & " " & time$)
      print #99, "-------------"
      Print #99, "   "

       NumItems = dmAPIGet("values,s0," & JobId & ",method_arguments")
       For I = 0 To Val(NumItems) - 1
          TheString = dmAPIGet("get,s0," & JobId & ",method_arguments[" & str$(I) & "]")
          tempBadArgs% = 0
         For J = 1 To numArgs
           If validArgs(J) = Item$(TheString, 1, 1, " ") Then

‘PRINT THE VALUES THAT THE CODE IS COMPARING JOB ARGUMENTS VS TOOLSET.INI

              Print #99, "ResultFile value """ & validArgs(J) & """=" & Item$(TheString, 1, 1, " ")
              print #99, "-------------"
              tempBadArgs% = 1
              Exit For
           End If
         Next J
         If tempBadArgs% = 0 Then
           CheckForBadArgs = 0

‘IF THE ARGUMENT IS BAD WE WILL PRINT WHICH ONE IS FAILING

           Print #2, TheString & " is a bad argument."
           Print #99, "Bad argument """ & TheString & """ "
           print #99, "-------------"
         End If
       Next I

‘CLOSING THE DEBUG OUTPUT

  close #99
End Function
```

Then we run the job and the “clean.txt” file (debug file) will be created in:

$DM_HOME/bin

Once you open the file you will see the following output:

----------Debug file output with bad argument----------

Start of output

Start of log:6/25/2003 10:05:15

ResultFile value "-window_interval"=-window_interval

ResultFile value "-queueperson"=-queueperson

ResultFile value "-clean_content"=-clean_content

ResultFile value "-clean_note"=-clean_note

ResultFile value "-clean_acl"=-clean_acl

ResultFile value "-clean_now"=-clean_now

Bad argument "-custom_predicate"

----------Debug file output with bad argument----------

As you can see there is an argument in the job that is not in the toolset.ini file, if you remove the argument and run the job again the debug output file should look like this:

----------Debug file output with good arguments----------

Start of output

Start of log:6/25/2003 09:55:44

ResultFile value "-window_interval"=-window_interval

ResultFile value "-queueperson"=-queueperson

ResultFile value "-clean_content"=-clean_content

ResultFile value "-clean_note"=-clean_note

ResultFile value "-clean_acl"=-clean_acl

ResultFile value "-clean_now"=-clean_now

----------Debug file output with good arguments----------

Support Notes

Note:1518

How can I manually install the DocPage Server Administration tools?

Run the script file "dm_apply_admin_install" located for UNIX in the file system path "$DM_HOME/install."

For NT, run the script "toolset.ebs" found in the path " $DOCUMENTUM/product/3.1/install/admin." Launch the script at the DOS command prompt using the following syntax:

"dmbasic -eToolSetup -fc:\documentum\product\3.1\install\admin\toolset.ebs -p<docbase name> c:\documentum\product\3.1\install\admin DM_RDBMS_SERVER_NAME"

(The assumption that the Documentum Server Installation is on C drive.)

On Unix, execute the following script:

$DM_HOME/install/dm_apply_admin_install script.

To run this script, be sure you are logged into the OS as the docbase_owner account.

In some cases this will fail due to the group "admingroup" and the ACL "dm_acl_superusers" does not exist. If this happens, you can create the group "admingroup" and add the docbase owner account as a member. You can also create the ACL "dm_acl_superusers" with the permissions "OWNER-DELETE, WORLD-NONE, ADMINGROUP - DELETE." This ACL should be a system ACL created by the docbase owner.

Note: 16661

How do find if a method is running?

Execute the following script:

```
select r_lock_date, r_lock_owner from dm_method
where
object_name = '_method_name_';
```

If the r_lock_date and r_lock_owner are NULL, then the job is not running.
Please refer to SN 15789, 8404 for information on how to unlock.

Note: 15568

What method_verb should I use when creating a server method which uses Java?

The method verb to use when running a Java server method (a dm_method) will depend upon the platform of your eContent Server (eCS).

If your eContent Server is running Sun Java (which is the case with Solaris and HPUX machines), then the method verb will be:

java myClass

If your eContent Server is running Jview, the Microsoft Command-line Loader for Java, (as is usually the case with and eCS running on NT), then the method verb will be:

jview myClass

Note that the .class extension is always left off. Although your compiled class is myClass.class, the method verb will reference just myClass.

These examples assume that you have placed your .class file in the product bin directory on the e-content server.

If you are curious about method_verb settings used by java methods in your e-content server, you can run a query like:

select * from dm_method where method_verb like '%jview%'

Then look in the product bin directory on the e-content server and you will see the .class files corresponding to these methods.

For more on Java server methods, see Support Notes 5999, 14627, and 6754 and 9138.

Note: 1682

How do you pass parameters to a method? or Why is my LAUNCH TRACE showing <parameter argument>: No such file or directory?

There are two ways of passing parameters to a method. The first is to specify the -p flag as part of the argument string in the APPLY,DO_METHOD command. This forces the -p flag preceding the arguments. The apply command will look something like the following:
apply,c,NULL,DO_METHOD,METHOD,S,test_julie_method,SAVE_RESULTS,B,T,ARGUMENTS,S,'-pJUNK'

The other way is valid only if the method is a dmbasic method. You need to specify 'dmbasic' as the method_type. This causes the server to append the magic '--' string after the method verb but before the arguments so that dmbasic can recognize the arguments passed via APPLY,DO_METHOD if you do not specify the -p flag.

Note: 22613

How can I move method contents to the file system?

You can move Documentum method contents to the file system by copying them to a folder and then update your method verb.

Followed by:

getfile,c,<id_returned>,<path to documentum/product/<version>/bin directory including method file name if you are on the eContent server>

for example:

getfile,c,100007d28000695c,c:\documentum\product\<version>\bin\<method.ebs>

NOTE: You can do this via Documentum Administrator, but when you do a getfile in API the method content will be stored on the machine where Documentum Administrator is located.

If you are not going the getfile from the e-Content Server you'll need to move the files over to the Documentum\product\<version>\bin directory (or where ever you plan to store the method content).

Then log into Documentum Administrator and click on the methods link. Select your method and click the edit button. Add the -f flag and the method name to the method_verb. For example, your method_verb probably looks something like:

.\dmbasic -eEntry_point

I'd like you to change it to look like:

.\dmbasic -f<method_name> -eEntry_point

If you want to put your method content in a directory other than Documentum\product\<version>\bin you'll need to add the path to the file along with the file name after the -f flag.

For example:

.\dmbasic -fc:\temp\<method_name> -eEntry_point

Do this for each method that is stored in the docbase.

Next in the DQL area in Documentum Administrator type the following query:

update dm_method object
set use_method_content = 0
where object_name = '<method_name>'

This tells the server to look on the file system for the method content instead of in the docbase. You'll need to do this for each method you plan to store on the file system.

Finally, delete the content in the $DOCUMENTUM/share/data/common/<hex docbase id>/80/00/00/07 directory and restart the eContent Server.

Runtime Error Messages

This section contains listings of all the runtime errors. It is divided into two subsections, the first describing errors messages compatible with "standard" Basic as implemented by Microsoft Visual Basic and the second describing error messages specific to Docbasic.

A few error messages contain placeholders, which get replaced by the runtime when forming the completed runtime error message. These placeholders appear in the following list as the italicized word placeholder.

Visual Basic Compatible Error Messages

Error Number	Error Message
3	Return without GoSub
5	Illegal procedure call
6	Overflow
7	Out of memory
9	Subscript out of range
10	This array is fixed or temporarily locked
11	Division by zero
13	Type mismatch
14	Out of string space
19	No Resume
20	Resume without error
26	Dialog needs End Dialog or push button
28	Out of stack space
35	Sub or Function not defined
48	Error in loading DLL
49	Bad DLL calling convention
51	Internal error
52	Bad file name or number
53	File not found
54	Bad file mode
55	File already open
57	Device I/O error
58	File already exists
59	Bad record length
61	Disk full
62	Input past end of file
63	Bad record number
64	Bad file name
67	Too many files

Error Number	Error Message
68	Device unavailable
70	Permission denied
71	Disk not ready
74	Can't rename with different drive
75	Path/File access error
76	Path not found
91	Object variable or With block variable not set
93	Invalid pattern string
94	Invalid use of Null
139	Only one user dialog may be up at any time
140	Dialog control identifier does not match any current control
141	The placeholder statement is not available on this dialog control type
163	This statement can only be used when a user dialog is active
260	No timer available
281	No more DDE channels
282	No foreign application responded to a DDE initiate
283	Multiple applications responded to a DDE initiate
285	Foreign application won't perform DDE method or operation
286	Timeout while waiting for DDE response
287	User pressed Escape key during DDE operation
288	Destination is busy
289	Data not provided in DDE operation
290	Data in wrong format
291	Foreign application quit
292	DDE conversation closed or changed
295	Message queue filled; DDE message lost
298	DDE requires ddeml.dll
429	OLE Automation server can't create object
430	Class doesn't support OLE Automation
431	OLE Automation server cannot load file
432	File name or class name not found during OLE Automation operation
433	OLE Automation object does not exist
434	Access to OLE Automation object denied
435	OLE initialization error
436	OLE Automation method returned unsupported type
437	OLE Automation method did not return a value
438	Object doesn't support this property or method placeholder
439	OLE Automation argument type mismatch placeholder
440	OLE Automation error placeholder
443	OLE Automation Object does not have a default value
452	Invalid ordinal
460	Invalid Clipboard format
520	Can't empty clipboard
521	Can't open clipboard

Error Number	Error Message
600	Set value not allowed on collections
601	Get value not allowed on collections

Docbasic-Specific Error Messages

Error Number	Error Message
800	Incorrect Windows version
801	Too many dimensions
802	Can't find window
803	Can't find menu item
804	Another queue is being flushed
805	Can't find control
806	Bad channel number
807	Requested data not available
808	Can't create pop-up menu
809	Message box canceled
810	Command failed
816	Queue overflow
821	Can't write to ini file
822	Can't read from ini file
823	Can't copy file onto itself
824	OLE Automation unknown object name
825	Can't redimension a fixed array
826	Can't load and initialize extension
827	Can't find extension
828	Unsupported function or statement
830	OLE Automation Lbound or Ubound on non-Array value

Compiler Error Messages

The following table contains a list of all the errors generated by the Docbasic compiler. With some errors, the compiler changes placeholders within the error to text from the procedure being compiled. These placeholders are represented in this table by the word placeholder.

Error Number	Error Message
1	Variable Required - Can't assign to this expression
2	Letter range must be in ascending order
3	Redefinition of default type
4	Out of memory, too many variables defined
5	Type-character doesn't match defined type
6	Expression too complex
7	Cannot assign whole array
8	Assignment variable and expression are different types
9	Identifier is not an array variable
10	Array type mismatch in parameter
11	Array type expected for parameter
12	Array type unexpected for parameter
13	Integer expression expected for an array index
14	Integer expression expected
15	String expression expected
16	Long expression expected
17	Invalid dialog element
18	Left of "." must be an object, structure, or dialog
19	Invalid string operator
20	Can't apply operator to array type
21	Operator type mismatch
22	"placeholder" is not a variable
23	"placeholder" is not a array variable or a function
24	Unknown placeholder "placeholder"
25	Out of memory
26	placeholder: Too many parameters encountered
27	placeholder: Missing parameter(s)
28	placeholder: Type mismatch in parameter placeholder
29	Missing label "placeholder"
30	Too many nested statements
31	Encountered new-line in string
32	Overflow in decimal value
33	Overflow in hex value
34	Overflow in octal value
35	Expression is not constant
36	Function name not explicitly typed or missing type-character

Error Number	Error Message
37	No type-characters allowed on parameters with explicit type
38	Missing type for variable
39	Can’t pass an array by value
40	"placeholder" is already declared as a parameter
41	Variable name used as label name
42	Duplicate label
43	Not inside a function
44	Not inside a sub
45	Long expression expected
46	Can't assign to function
47	Identifier is already a variable
48	Unknown type
49	Variable is not an array type
50	Can't redimension an array to a different type
51	Identifier is not a string array variable
52	0 expected
53	Parameter 1: string expected
54	Parameter 2: string expected
55	Integer expression expected for file number
56	placeholder is not a method of the object
57	placeholder is not a property of the object
58	Expecting 0 or 1
59	Boolean expression expected
60	Numeric expression expected
61	Numeric type FOR variable expected
62	For...Next variable mismatch
63	Out of string storage space
64	Out of identifier storage space
65	Internal error 1
66	Maximum line length exceeded
67	Internal error 3
68	Division by zero
69	Overflow in expression
70	Floating-point expression expected
71	Floating-point expression expected
72	Invalid floating-point operator
73	Math error
74	Single character expected
75	Subroutine identifier can't have a type-declaration character
76	Procedure is too large to be compiled
77	Variable type expected
78	Can't evaluate expression

Error Number	Error Message
79	Can't assign to user or dialog type variable
80	Maximum string length exceeded
81	Identifier name already in use as another type
82	Left of "." must be an object, structure, or dialog
83	Operator result on variant type is ambiguous
84	Operator cannot be used on an object
85	placeholder is not a property or method of the object
86	Type-character not allowed on label
87	Type-character mismatch on routine placeholder
88	Destination name is already a constant
89	Can't assign to constant
90	Error in format of compiler extensions
91	Identifier too long
92	Expecting string or structure expression
93	Can't assign to expression
94	Dialog and Object types are not supported in this context
95	Array expression not supported as parameter
96	Dialogs, objects, and structures expressions are not supported as a parameter
97	Invalid numeric operator
98	Invalid structure element name following "."
99	Access value can't be used with specified mode
100	Use the Set statement to assign objects
101	Invalid operator for object
102	Can't LSet a type with a variable-length string
103	Syntax error
104	Placeholder is not a method of the object
105	No members defined
106	Duplicate type member
107	Set is for object assignments
108	Type-character mismatch on variable
109	Bad octal number
110	Bad number
111	End-of-procedure encountered in comment
112	Misplaced line continuation
113	Invalid escape sequence
114	Missing End Inline
115	Statement expected
116	ByRef argument mismatch
117	Integer overflow
118	Long overflow
119	Single overflow

Error Number	Error Message
120	Double overflow
121	Currency overflow
122	Optional argument must be Variant
123	Parameter must be optional
124	Parameter is not optional
125	Expected: Lib
126	Illegal external function return type
127	Illegal function return type
128	Variable not defined
129	No default property for the object
130	The object does not have an assignable default property
131	Parameters cannot be fixed length strings
132	Invalid length for a fixed length string
133	Return type is different from a prior declaration
134	Private variable too large. Storage space exceeded
135	Public variables too large. Storage space exceeded

About Documentum
Documentum provides enterprise content management software solutions to more than 1,400 of the largest businesses in the world. The company brings intelligence and automation to the creation, management, personalization, and distribution of vast quantities and types of content — documents, Web pages, XML files, rich media — in one common content platform and repository. The Documentum platform makes it possible for companies to distribute content globally across all internal and external systems, applications, and user communities — all while maintaining brand and user experience. We put content to work by delivering the right content to the right user at the right time.

For more information about Documentum, visit www.documentum.com or call 800.607.9546 (outside the U.S.: +1.925.600.6754)

Documentum, Inc.
6801 Koll Center Parkway
Pleasanton, CA 94566-7047
phone (925) 600-6800
fax (925) 600-6850
www.documentum.com