

New Features in Panther 5.30



Windows, Character Mode, Motif

38146 – [Trace Log Feature](#), ability to trace events such as JPL execution, database calls. Event information can be written to a trace file as they occur.

38167 – File type is pre-set in the Library Table of Contents window if Save Preferences on Exit is selected.

38060 - JDK 6 Support.

38062 – Native 64bit Windows port (Win).

38069 – Tooltips now appears for toolbox (Linux).

38093 – Improved link parameters.

38096 – Support for all 6 arguments in the installed prototypes functions.

38101 - Ability to center sm_message_box dialog in MDI frame/screen (Win, Motif).

38106 – Increase Max Size of JPL Arrays to 1,052,688 for a variable of 255 characters in length.

38107 - Access to Panther setup variables from smvars files using Property API .
i.e. @app->java_use, or @app->smpath etc.

38109 – Visual Studio 2010 support (Win).

38229 - Support for Operating System Authentication in Oracle

Report Writer

38138 – Support for tiff image using PDF Driver.

Databases

38094 – Short Escape sequence support for ODBC is now the default.

38147 - Sybase 15 Support.

38088 - Support access to Oracle CLOB (character large object) columns. See the Library Function Documentation section below.

Library Functions

38132 - sm_mw_DismissIntroPixmap() allows one to close IntroPixmap window.

38183 - sm_d_jplpublic and sm_n_jplpublic, load JPL modules from memory .

38210 - sm_com_attach() ability to attach to an active COM object's process.

38808 - dm_ora_clob_update()update a CLOB (character large object) column after selecting the column. See the Library Function Documentation section below.

Panther 64-bit for Windows

1. Panther 64-bit for Windows provides 64-bit libraries and executables to be used on 64-bit Windows operating systems. Although Panther 32-bit for Windows may be used on 64-bit Windows operating systems, Panther 64-bit must be used when linking with third party 64-bit libraries. For example, use Panther 64-bit if 64-bit Oracle Client libraries is required. The only 32-bit components that may be used with Panther 64-bit are COM and ActiveX components.
2. All file names that had the suffix "32" for the Panther 32-bit are replaced by file names using "64" as their suffix for Panther 64-bit. For example, "pro15w64.ini" is used instead of "pro15w32.ini." This is a change from Panther 5.20 64-bit. (Note that Panther 64-bit for Windows had a limited release as a late addition to Panther 5.20.)
3. A native MS SQL driver is not supported in Panther 64-bit. Use the ODBC driver instead.
4. Panther 64-bit can create containers for existing ActiveX components as well as access COM objects, but cannot be used to build COM Components. Panther 64-bit does not support Microsoft Transaction Server (MTS) or COM+.
5. The Panther Editor and runtime for Panther 64-bit require 'msvcr80.dll', a Microsoft Visual C++ 2005 redistributable file, as well as its manifest, 'Microsoft.VC80.CRT.manifest'. These files are installed in the 'util' directory, and should be redistributed with Panther 64-bit applications. Install these two files in the same directory as any executables for Panther 64-bit end-user applications.

Library Function Documentation

`dm_ora_clob_update()`

Update a CLOB (character large object) column after selecting the column

```
retcode = dm_ora_clob_update (cursor_name, column_number, field_name);
    int retcode;                returns 0 or an error code
    char *cursor_name;         name of the cursor
    int column_number;        number of column in select list
    char *field_name;         name of the wordwrap field holding the
                             data to write to the CLOB
```

sm_trace

Creates event trace and dump files

```
int sm_trace(char *commands);
```

commands Command to execute

Returns: 0 if there are no errors.
 Otherwise, one of these error codes:

SM_EQUALS	No equal sign following a DUMPFIL
SM_FORMAT	non-alphabetic character in a command.
SM_MALLOC	a call to the malloc function has failed.
SM_MISSARGS	command is the null pointer; null string or just whitespace.
SM_NOFILE	unable to open the file in a DUMPFIL or TRACEFILE command.
SM_NOT_LOADED	DUMP command when DUMPFIL is not specified.
SM_NUMBER	DUMP command when FRAMES is zero or the FRAMES command value is not one or more digits.
SM_QUOTE	DUMPFIL or TRACEFILE value starts with a quote character but terminating quote not found.
SM_SYNTAX	character after the closing quote in a DUMPFIL or TRACEFILE is not a NULL or whitespace.
SM_VERB_UNKNOWN	command is not valid.

Description :

sm_trace can create trace and dump files containing information about Panther events. The events to report in these files can be controlled. The commands parameter is a series of tokens that control operations. Most tokens can be prefixed with NO to turn off their normal effect. Case is ignored but tokens will be capitalized in this document.

Trace files are written as events occur. The following commands control trace file operations:

TRACEFILE=	File to write event information to. The file name should be quoted if it contains spaces. Use the null string "" to stop tracing and close the current TRACEFILE.
------------	---

TRACE	Used to control whether event information is to be written to TRACEFILE. Default is TRACE.
-------	--

Dump files are written when the DUMP command is executed. The number of events to store for reporting is controlled by the FRAMES= command. The following tokens control dump files:

DUMPFIL=	File to write information to. The file name should be quoted if it contains spaces.
DUMP	Used to cause information is to be written to DUMPFIL=.
FRAMES=	Specifies how many events are to be stored for writing by the DUMP command. 500 is the default.

The following tokens control how files are opened:

OVERWRITE	Specifies that existing files should be overwritten. It is the default.
APPEND	Specifies that existing files should be appended to.

The following tokens control which events are to be logged:

CSTR	Control String events
DBI	Database (DBi) events
FIELD	Field entry, exit and validation events
FUNCTION	Installed function call and return events
GRID	Grid and Grid Row entry, exit and validation events
GROUP	Group entry, exit and validation events
JPL	JPL execution events
KEY	Key being taken off the key queue events
LDB	LDB (Local Data Block) events
RW	Report Writer (RW) events
SCREEN	Screen entry, exit and expose events
TM	Transaction Manager (TM) events
ALL	All of the above events.
CORE	All of the above events except for DBI, RW and TM.
NONE	The same as NOALL.
PARMS	Causes parameters to be displayed for calls to functions, including JPL functions.

Example:

Use the setup variable `SMTRACE` in the `smvars` file to pass parameters to `sm_trace` at startup time. As an example:

```
SMTRACE          = TRACE NORW TRACEFILE="c:\test\PantherTrace"
```

The above command turns tracing on, ignores Report Writer events; and writes the trace to "c:\test\Panther Trace".