

Panther

New Features

Release 5.00

Prolifics[®]

This software manual is documentation for Panther™ 5. It is as accurate as possible at this time; however, both this manual and Panther itself are subject to revision.

Prolifics and JAM are registered trademarks and Panther and JAM/TPi are trademarks of JYACC, Inc.

Adobe, Acrobat, and PostScript are registered trademarks and Acrobat Reader a trademark of Adobe Systems Incorporated.

BEA and BEA Tuxedo are registered trademarks of BEA Systems, Inc.

FLEX*lm* is a registered trademark of GLOBEtrouter Software, Inc.

IBM, AIX, DB2, RISC System/6000, and VisualAge are registered trademarks and WebSphere is a trademark of International Business Machines Corporation.

Informix and C-ISAM are registered trademarks of Informix Corporation.

INGRES is a registered trademark of Computer Associates International, Inc.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Microsoft, MS-DOS, ActiveX, Visual C++, Visual J++, Windows, Win32 and Windows NT are registered trademarks and Microsoft Windows 95, Authenticode, Microsoft Transaction Server, Microsoft Access, Microsoft Internet Explorer, Microsoft Internet Information Server, Microsoft Management Console, and Microsoft Open Database Connectivity are trademarks of Microsoft Corporation in the United States and/or other countries.

Motif is a registered trademark of The Open Group in the United States and other countries.

Netscape, Netscape FastTrack Server, and Netscape Navigator are registered trademarks of Netscape Communications Corporation in the United States and other countries.

Oracle and SQL*Net are registered trademarks and Oracle7, Oracle8, Oracle8i, PL/SQL, Pro*C, Rdb7, and Rdb8 are trademarks of Oracle Corporation.

Rational Rose is a registered trademark of Rational Software Corporation in the United States and in other countries.

Sun, SunOS, and Solaris are trademarks and Sun Workstation is a registered trademark of Sun Microsystems, Inc. in the United States and other countries.

Sybase is a registered trademark and Client-Library, DB-Library, and SQL Server are trademarks of Sybase, Inc.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product names mentioned in this manual may be trademarks or registered trademarks of their respective owners, and are used for identification purposes only.

Send suggestions and comments regarding this document to:

Technical Publications Manager

Prolifics

116 John Street

New York, NY 10038

<http://www.prolifics.com>

support@prolifics.com

(212) 267-7722

© 1997–2004 JYACC, Inc.

All rights reserved.

Printed in USA.

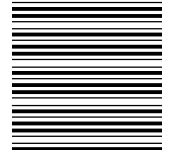


Table of Contents

About this Guide	v
Organization of this Guide	v
Conventions	v
Text Conventions	v
Keyboard Conventions	vi
Panther Documentation	vi
Chapter 1 New in Panther 4	1
Program Startup	1
Editor	1
Menu Options	2
Screen Name Extensions	3
Tab Control	4
Non-modal Window for Editing JPL, JavaScript, and VBScript	4
Date Formats for Year 2000 Compliance	4
Docking Toolbars	5
Wizard Column Selection	5
Grids and Arrays	5
ActiveX Controls	6
Properties	6
New Properties	6

Property Changes	14
Property Window Changes	15
Application Property Syntax	16
Readable Transaction Properties	16
Writable Transaction Manager Properties	16
JPL Programming	16
Declaring Variables	16
Sending and Receiving Data	16
Variable Assignments	17
Forwarding Messages	17
Component-based Applications	17
Application Property Syntax	17
Library Functions	17
New Functions	17
Changed Functions	21
Discontinued Functions	22
Additional Flags for Widget Functions	24
Development Environment	24
Java Interface	24
Team Development	24
QA Partner Interface	25
Internal File Locking Available on Windows NT	25
COM Components in Axview	25
Opening Library Files in Windows	25
MSVC Project Files	25
Text Selection Keys	26
Three-Tier Application Development	26
Configuration	26
Utilities	28
New Utilities	28
COM/MTS Utilities	28
WebSphere Utilities	28
Changed Utilities	28
Discontinued Utilities	29
Database Interface	30
Improved SQL Processing	30
Specifying Variables in DECLARE CONNECTION	30
Importing Database Tables	31
JDB Databases	31

Database Drivers	31
Library Functions	32
Transaction Manager	32
Transaction Manager Common Model	33
Releasing Database Cursors in the Transaction Manager	34
Calling Transaction Event Functions	34
Checking the Size of the Select Set	34
Support for Outer Joins	35
Generating SQL Statements	35
SQL Statement Handling	35
Writable TM Properties and Default Transaction Name	36
Web Application Development	36
Web Setup Manager	36
Web Initialization Files	36
Caching Application State	37
Web Entry Processing	38
HTML Template Processing	38
Web-specific Properties	38
Determining Mouse Location	40
Requester Executables	40
Java Servlets	40
Windows NT Service	41
Widget Positioning in Web Applications	41
Web Wizard Defaults	41
Naming Conventions	41
Reports	41
Report Utilities	41
Three-tier Development	41
JetNet and Tuxedo	42
Open Middleware Connectivity	43
Panther WebSphere Applications	44
Documentation	44
Online Documentation Changes and Corrections	45
Quick Reference Changes and Corrections	46
Chapter 2 New in Panther 4.2	49
Chapter 3 New in Panther 4.25	51
Chapter 4 New in Panther 4.5	53
Chapter 5 New in Panther 5	55



About this Guide

Organization of this Guide

This guide describes what is new and different in the Panther products. For users updating from JAM, refer instead to the *Upgrade Guide* for information on changes from the JAM products.

If you have never used any of the Panther, Prolifics or JAM products, you should start with the Introducing Panther in *Getting Started* and Building a Panther Application in *Application Development Guide*.

Conventions

The following typographical and terminological conventions are used in this guide:

Text Conventions

`expression`

Monospace (fixed-spaced) text is used to indicate:

- Code examples.

- Words you're instructed to type exactly as indicated.
- Filenames, directories, library functions, and utilities.
- Error and status messages.

KEYWORDS

Uppercase, fixed-space font is used to indicate:

- SQL keywords.
- Mnemonics or Prolifics constants.

numeric_value

Italicized helvetica is used to indicate placeholders for information you supply.

[*option_list*]

Items inside square brackets are optional.

{*x* | *y*}

One of the items listed inside curly braces needs to be selected.

x ...

Ellipses indicate that you can specify one or more items, or that an element can be repeated.

new terms

Italicized text is used:

- To indicate defined terms when used for the first time in the guide.
- Occasionally for emphasis.

Keyboard Conventions

XMIT

Prolifics logical keys are indicated with uppercase characters.

Alt+A

Physical keys are indicated with initial capitalization, and keys that you press simultaneously are connected with a plus sign.

Panther Documentation

The Panther documentation set is available online as part of the Panther distribution and on a separate Windows documentation CD. It includes the following guides and reference material, as illustrated in Figure 1:

Installation — Instructions for installing software and licensing for Panther and its Prolifics products in the following books: *JetNet Installation*, *TUXEDO Installation*, *2-Tier/COM/MTS Installation*, *Panther for IBM WebSphere Installation*, and *2-Tier Installation (UNIX)*.

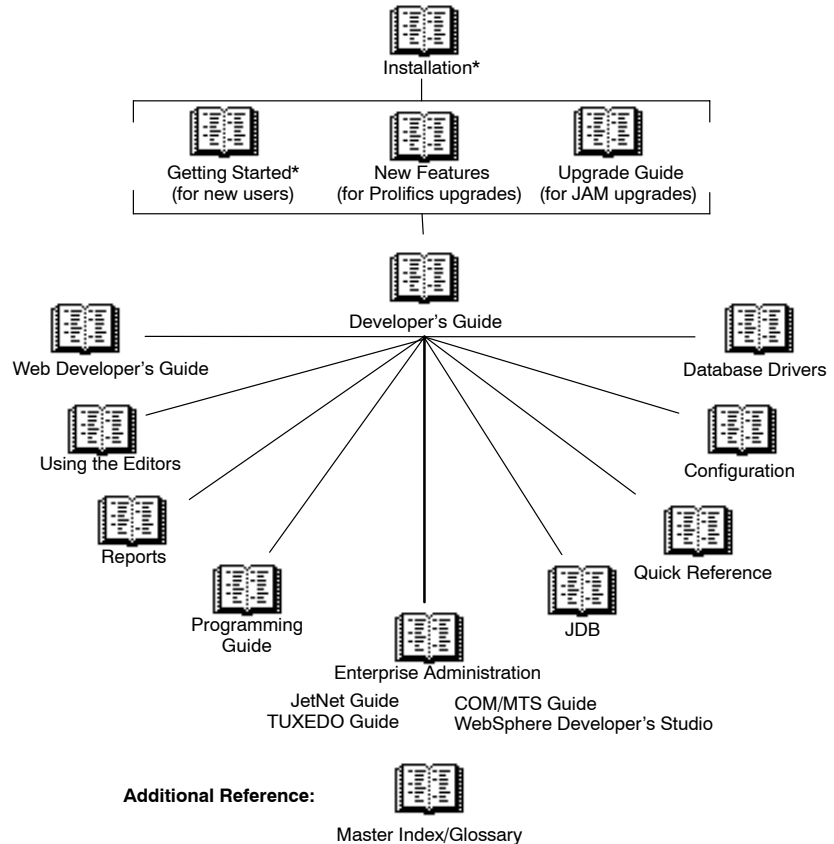


Figure 1. A suggested reading path for the Panther documentation set. An asterisk (*) indicates that there are multiple versions of the book for different environments.

New Features Guide — Information about new features in Panther.

Upgrade Guide — Information for upgrading from JAM to Panther.

Getting Started — Introduction to the Panther framework, its Prolifics products, and the tutorial, with step-by-step instructions that help you get started with understanding, setting up and developing either a Panther application in a two-tier, JetNet or Tuxedo environment.

Developer's Guide — Introduction and process of how to use Prolifics products to build n -tier Panther applications. As the major developer's manual, it includes information for each step of the development path.

Web Development Guide — Introduction and process of how to build and deploy a Panther web application.

Using the Editors — Instructions for using the Panther graphical authoring environment—the editor, screen and report wizards, JIF editor, menu bar editor, and styles editor—to build two- and three-tier applications.

Programming Guide— Describes JPL, Panther’s scripting language, and Panther’s Java programming interface. Also includes an alphabetical listings of JPL commands, built-in functions, Panther’s C library functions, transaction manager commands, and database commands and variables.

Reports — Instructions for using Panther’s report generation utility to build, modify and run reports.

JDB — Instructions for using JDB, Panther’s prototyping database, and JISQL, the interactive SQL editor for JDB. Also describes how to build SQL statements and the SQL commands available in JDB.

Database Drivers — Instructions for using Panther’s database drivers. Each driver has its own section containing the information specific to that relational database engine.

Configuration — Instructions for configuring Panther on various platforms and to your preferences. Also includes information on GUI resource and initialization files.

JetNet Guide — Instructions for configuring and monitoring your three-tier Panther JetNet application.

TUXEDO Guide — Instructions for configuring and monitoring your three-tier Panther TUXEDO application.

COM/MTS Guide — Instructions for building COM components in the Panther editor and for deploying components in your Panther application using COM, DCOM, and MTS.

WebSphere Developer’s Studio — Instructions for building Enterprise JavaBeans to be deployed in the WebSphere Application Server.

Collateral Documentation

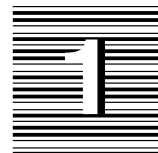
The following information is also provided with your Panther installation:

- Online README file.

Additional Help

Contact Prolifics for more information on the following services:

- Product Support
- Consulting Services
- Educational Services



New in Panther 4

With Panther, you are provided an open development framework for using and building Prolifics software components as well as interfacing to other COM components and Enterprise JavaBeans.

This chapter lists features that are new or changed since Prolifics 2.5. The Panther 4.2 changes have also been extracted in a separate list (page 49), the Panther 4.25 changes (page 51).

For customers upgrading from JAM, these features are also listed in the *Upgrade Guide* (refer to page 1 in the *Upgrade Guide*).

Program Startup

By default, Java is initialized on program startup. You can change this initialization using the new behavior variable `JAVA_USE`.

Windows clients having an old version of the Java DLLs display an error message, `Java Not Supported`. To update the Java DLLs, run the executable in `$SMBASE\jvm`.

Editor

Refer to page 6 for descriptions of new properties and information about changes in the Properties window.

Menu Options

File menu

The New, Open and Save options have the following new options:

- Service Component — Creates or opens a service component for use in distributed application processing.
- Java — Opens an external editor for writing or editing a Java file.

The Import⇒Database Objects option is now on the Tools menu.

In JetNet and Tuxedo executables, the Open/Close Middleware Connection options replace the Open/Close Request Broker options.

Edit menu

Includes the following new options, as well as corresponding toolbar buttons, which are available when a JPL library module is open or when the screen- or report-level JPL Procedures, JavaScript, or VBScript properties are being written/edited:

- Insert From Library — Includes a JPL , JavaScript, or VBScript file from another open library.
- Read File — Includes a JPL, JavaScript, or VBScript file from disk.
- External Editor — Invokes the editor you specify (via the `SMEDITOR` variable).

Create menu

A new widget type, tab controls, is available for Windows 32-bit applications; refer to page 4 for more information.

View menu

Includes the following new option:

- Component Interface – For Panther COM and Panther WebSphere executables, specify the properties, methods and settings for service components.

Options menu

Includes the following new options:

- Direct to External Editor — When active, invokes your preferred text editor (specified via the `SMEDITOR` variable) when you open the JPL, JavaScript, or

VBScript window; or edit properties that allow input of multiple lines of text, such as the Control Strings property and the Initial Text property (when it is associated with a widget having an array size greater than one).

- Configure Toolbars — For Windows 32-bit executables, allows you to specify which toolbars to display in the editor.
- Editor Tabs — Works in conjunction with the Direct to External Editor menu options to allow you to specify the number of spaces that defines a TAB character.
- Service Alias — For developing JetNet and Tuxedo applications, specify a user identifier to use when testing services. For more information, refer to page 67 in the *Application Development Guide*.
- Reload Java Classes — When active, reloads the Java classes when entering test mode or exiting the editor.

Tools menu

This new menu bar item contains the Import Database Objects that was located on the File menu in previous releases, and gives editor access to the styles editor, menu bar editor, and JIF editor (in JetNet and Tuxedo executables). There are also the following new options in Panther:

- Generate TM SQL — For the current screen, writes the SQL statements that the transaction manager generates for the screen to a file. These SQL statements could then be used to construct stored procedures or invoke DBMS QUERY or DBMS RUN directly.
- Generate Component — For COM components, generate a type lib file after changing the component interface without having to save the Panther service component. For Enterprise JavaBeans, generate and/or compile the bean's Java files.
- Compile Java — Compiles the specified Java class.
- IBM VisualAge for Java — (WebSphere only) Starts IBM's Visual Age for Java program.
- IBM WebSphere Administrative Console — (WebSphere only) Starts IBM's WebSphere Administrative Console where you install and deploy Enterprise JavaBeans.

Screen Name Extensions

No default value is set for the application variable `SMFEXTENSION`. If you do not explicitly set this variable in a setup file or the environment, Prolifics no longer

adds an extension to names during file searches; and all filenames must be fully qualified; for example, supplied a screen name of `myscreen`, Prolifics searches only for `myscreen`, not for `myscreen.ext`.

The recommended file extension for binary screen files is `.scr`. For binary report files, the recommended extension is `.rpt`. The file extension for temporary file names has been changed from `.jam` to `.pro`.

Tab Control

For Windows 32-bit applications, the tab control allows widgets to be grouped onto individual display “cards.” The tab control or “deck” contains a series of tab cards; these cards are accessed by means of index tabs, which are analogous to the dividers in a notebook or the labels on a group of file folders. By means of such a widget, you can define multiple cards for the same area of a window—grouping common information for the convenience of users. New logical keys, `NCARD` and `PCARD`, move to the next card and previous card respectively. For more information, refer to page 313 in the *Using the Editors*.

Non-modal Window for Editing JPL, JavaScript, and VBScript

The JPL Program Text window, JavaScript window, and VBScript window are no longer modal when writing screen- or report-level JPL procedures or JavaScript/VBScript functions. This allows you to edit multiple files as well as move freely between script files and the editor workspace. The buttons have also changed for these windows. `OK`, `Apply`, `Cancel`, and `Editor` are the current button choices at screen- or report-level.

The edit window can also be invoked using `Edit⇒External Editor` (or the toolbar button) or using `Options⇒Direct to External Editor` to invoke the editor directly.

The Edit menu options associated with the editor workspace can be used when editing your script files. For example, the Edit menu options, `Cut`, `Copy`, `Delete`, `Paste`, and `Select All` can be used. `Insert From Library` and `Read File` menu options (and toolbar buttons), formerly `Insert` and `Insert File` buttons on the JPL Program Text window, let you insert JPL from another library or read a file from disk into the text window, respectively.

Date Formats for Year 2000 Compliance

Two new date/time Format Type specifications, which display a four-digit year, were added to the Properties window (and message file):

```
MON/DATE/YR4 HR:MIN2
MON/DATE/YR4
```

These are associated with the `DEFAULT 3` and `DEFAULT 4` mnemonics, respectively. In addition, `DEFAULT 3` is set as the default type in the Properties window.

Alternatively, you can use the behavior variable, `DA_CENTBREAK`, to set the behavior for applications using two-digit years. The default value of `DA_CENTBREAK` is 50. Therefore, if the year setting is equal to or greater than 50, the year is processed as 19xx; if the year is less than 50, the year is processed as 20xx. You can change the setting of `DA_CENTBREAK` by setting its value in the `smvars` file or at runtime by using `sm_option`.

Docking Toolbars

Windows 32-bit executables have the option in the editor and in Windows applications of having toolbars dock to the MDI frame or float within the MDI frame. A new menu option, `Options⇒Configure Toolbars`, determines which toolbars are available in the editor. A new menu pixmap property, `Hot Pixmap`, controls the appearance of the item when a mouse moves over an active toolbar item.

In addition, pixmaps can be specified for the `Inactive Pixmap` property. In previous releases, an `Inactive Pixmap` was a grayed version of the `Active Pixmap` in Windows applications. That capability is still there, but you can also specify a separate inactive pixmap.

For each toolbar state that you want to indicate in your application—active, inactive and hot—you must supply a pixmap for each toolbar item. The size of the pixmaps for the entire toolbar is taken from the size of the first pixmap.

At runtime, application properties control the appearance and position of the toolbar; refer to page 123 in the *Application Development Guide*

Wizard Column Selection

When constructing screens in the screen wizard, columns defined as being `NOT NULL` in the database are automatically selected to be part of the screen and are designated with the number symbol (`#`).

Grids and Arrays

For grids, there are new properties and functions which control the amount of space between grid rows and which sort the data appearing in grids. Grid row

margins can be controlled throughout the application using the application property `default_row_margin` or on a single grid frame using the `row_margin` property.

For grids and arrays, there are new properties and functions which allow you to sort the data and control the behavior when the user clicks on the column. The `Column Click Action` property specifies the action to occur when the user clicks on a grid column setting. The `Sort Order` property specifies the sort order for the data; `sm_obj_sort` and `sm_obj_sort_auto` are the library functions that control data sorting. For more information on sorting data in grids, refer to page 308 in the *Using the Editors*.

ActiveX Controls

Runtime licensing is now supported for ActiveX controls. Enter the license information in the `Runtime License` property (under `ActiveX`).

Properties

New Properties

There are several new properties associated with widgets, screens, and applications. Some are settable via the Properties window and others are readable and/or writable only at runtime.

Before Image Rows (`bi_string[iter]`)

A widget, runtime-only, property. Provides access to the before image values of rows in the transaction manager. The *iter* specification lets you walk through the list of rows.

Card (`card`)

For widgets on tab cards, a runtime, read-only property returning the object id of the tab card of which the widget is a member.

Card Entry Function (`card_entry_func`)

For tab cards, under `Focus`, the name of the function to be called when the tab card is entered.

Card Exit Function (`card_exit_func`)

For tab cards, under `Focus`, the name of the function to be called when the tab card is exited.

Card Expose Function (`expose_function`)

For tab cards, under Focus, the name of the function to be called when the tab card is made the topmost card in the deck.

Card Hide Function (`hide_function`)

For tab cards, under Focus, the name of the function to be called when the tab card ceases to be the topmost card in the deck.

Card Number (`card_number`)

For tab cards, under Identity, specify the number location of the card in the deck.

Column Click Action (`column_click_action`)

For widgets in grids, under Format/Display, specify the action—sort or custom function—that occurs when a user clicks on the grid column header.

Column Click Function (`column_click_func`)

For widgets in grids, under Format/Display, specify the custom function to invoke when a user clicks on the grid column header. For this property to be available, Column Click Action must be set to Custom.

Conceal Tabs (`conceal_tabs`)

For tab decks, under Identity, determine whether the index tabs for the cards in the deck are visible.

Connection Pooling (`conn_pool_size`)

In Panther/WebSphere applications, specify the number of concurrent database connections.

Continue Function Name (`continue_func_name`)

For table views in two-tier applications, specify the function for handling CONTINUE operations in the transaction manager for the specified server/table view. The Select Handling property must be set to Function Name.

Count Result (`count_result`)

A table view, runtime-only, property. This readable/writable property holds the value returned from a count query (from a TM_SELECT_COUNT event), that is, the total number of rows in the result set. The value is examined to determine whether to query the user about proceeding with the normal SELECT statement.

Count Select (`count_select`)

A table view property, located under Transaction, takes a value of Yes or No. Instructs the transaction manager whether or not to count the number of rows in a

result set and compare it (stored in the server view's `count_result` property) to a specified threshold (Count Threshold property) value before actually fetching data. This property is readable and writable.

Current Component System (`current_component_system`)

A runtime-only property that instantiates the type of component system currently in use. Before creating any service components, set this property to:

- `PV_SERVER_COM` for COM components.
- `PV_SERVER_EJB` for Enterprise JavaBeans deployed under WebSphere Application Server.

Deck (`deck`)

For tab cards, a runtime-only, read-only property returning the object id of the tab deck of which the tab card is a member.

Default Link (`default_link`)

For Web applications, specify the URL location for this hyperlink. (This replaces the `link` property in previous releases.) If the property is specified for an array, it is the hyperlink location for every occurrence in the array. (See Item Link.)

Default Row Margin (`default_row_margin`)

Use this application property to control the grid row height if the Row Margin property is not set for the grid frame.

Default Transaction (`default_tran`)

A runtime-only, read-only screen property that provides the name of the default transaction manager transaction. This property always contains the name, even if the transaction is not currently open, and can be used to stop and then re-start the default transaction when making runtime property changes.

Delete Function Name (`del_func_name`)

For table views, specify the function for handling delete statements in the transaction manager for the specified table view. The Delete Handling property must be set to Function Name.

Delete Handling (`delete_handling`)

For table views, select the method for handling delete statements in the transaction manager for the specified table view: SQL Statement Generation (`PV_HANDLING_SQL`), Function Call (`PV_HANDLING_FUNC`), or Nothing (`PV_HANDLING_NOTHING`).

Deleted Rows (`di_string[iter]`)

A widget, runtime-only, property. Provides access to the values of deleted rows in the transaction manager. The *iter* specification lets you walk through the list of deleted rows. Use in conjunction with the `num_del_images` property.

Endsession (`endsession`)

For Windows applications, an application property which specifies the function to call which closes down the application when Windows sends the `WM_ENDSESSION` message.

HTML Max Loop (`html_max_loop`)

For HTML templates using condition processing, specify the number of loop iterations to perform before terminating the process. The default setting is 1000.

HTML Max Nest (`html_max_nest`)

For HTML templates using condition processing, specify the number of nesting levels. Each `if`, `while`, or `include` constitutes one level. The default setting is 20.

In Server (`in_server`)

An application property which specifies which server is in use for a service component: `PV_SERVER_COM`, `PV_SERVER_MTS` or `PV_SERVER_EJB`.

Insert/Delete Buttons property (`ins_del_buttons`)

For grid frames in Web applications, if set to Yes (default), Insert (Insert Above and Insert Below) and Delete buttons are generated in the HTML representation of the grid frame. If set to No, the buttons are not generated under any circumstances.

Insert Function Name (`ins_func_name`)

For table views, specify the function for handling insert statements in the transaction manager for the specified table view. The Insert Handling property must be set to Function Name.

Insert Handling (`insert_handling`)

For table views, select the method for handling insert statements in the transaction manager for the specified table view: SQL Statement Generation (`PV_HANDLING_SQL`), Function Call (`PV_HANDLING_FUNC`), or Nothing (`PV_HANDLING_NOTHING`).

Item Link (`item_link`)

For Web applications, specify the URL location for this hyperlink. If specified, this property overrides the value in the Default Link and Image Map properties. If the widget is an array, you can specify a different URL location for each occurrence.

Java Tag (`java_tag`)

Under Identity, specify the Java class implementing the event handler for this object (screen, service component, widget).

Join Type (`join_type`)

For link widgets, under Transaction, a subproperty of Type. When the link is identified as a server link, that is if the link's Type (`type`) property is set to Server (`PV_LNK_SERVER`), the Join Type property is available. It can be set to: Inner (`PV_INNER`) (default), Left Outer (`PV_LEFT_OUTER`), Right Outer (`PV_RIGHT_OUTER`), or Full Outer (`PV_FULL_OUTER`). This property lets you take advantage of SQL join facilities, whereby you can control the join operation of a SELECT statement that combines information from two database tables.

Max Bundles (`max_bundles`)

A runtime-only application property specifying the number of JPL bundles available for send and receive commands. It defaults to ten bundles (including the unnamed bundle) if unspecified.

Number of Cards (`number_of_cards`)

For a tab deck, a runtime-only, read-only property specifying the number of cards in a tab deck, including hidden cards.

Number of Columns (`num_columns`)

A read-only and runtime-only property associated with table view widgets. This property returns the number of columns belonging to a specific table view, or more specifically, the number of occurrences defined in the Columns (`columns`) property.

Number of Deleted Rows (`num_del_images`)

A widget, read-only and runtime-only, property that returns the number of deleted rows in the transaction manager.

OnMouseOut (`on_mouse_out`)

For Web applications, under Browser Events, this property lets you specify a JavaScript or VBScript function to execute when the mouse pointer leaves an area (in client-side image maps) or a link.

Previous Form (`previous_form`)

For Web applications, get the screen name as stored in the current cache file. Typically, this would be the name of the last screen that was accessed.

Primary Key Update (`primary_key_update`)

A runtime-only application property determining how primary key changes are processed in the transaction manager: whether the row is updated or whether it is deleted and then inserted.

Provider URL (`provider_url`)

For WebSphere applications, a runtime-only application property specifying the location of the WebSphere application server machine. If `SMPROVIDERURL` is set in the environment, the property is initially set to this value.

Queryendsession (`queryendsession`)

For Windows applications, an application property which specifies the function to call which prepares to close the application when Windows sends the `WM_QUER- YENDESESSION` message.

Radio Buttons (`radio_buttons`)

For grid frames in Web applications, if set to Yes (default), radio buttons are generated for each occurrence in the grid frame if one of the members is unprotected or if the grid's Stripe Current Row property is set to Yes. If set to No, buttons are never generated.

Regenerate SQL (`regenerate_ins_sql, regenerate_upd_sql`)

If the transaction manager generates SQL statements, as determined by the Method property, specify if the SQL statement should be regenerated for each row in the table.

Row Margin (`row_margin`)

For grid frames, under Geometry, adjust the space between the text and row dividers to control the row height.

Runtime License (`runtime_license`)

For ActiveX controls which support runtime licensing, if the Runtime License property exists, the control will be created using the license.

Save Function Name (`save_func_name`)

For table views, specify the function for handling `SAVE` operations in the transaction manager for the specified server/table view. The Delete Handling, Insert Handling, or Update Handling properties must be set to Function Name.

Screen Type (`screen_type`)

For screens and service components in distributed applications, a property under Identity which displays whether the screen object is a client screen or service component.

Scroll Buttons (`scroll_buttons`)

For grid frames in Web applications, if set to Yes (default), allows scroll buttons (Page Up, Page Down, Top, and Bottom) to appear in the HTML representation of the grid when the number of occurrences in the grid exceed the number of onscreen rows. If set to No, the scroll buttons are not generated under any circumstances.

Select Function Name (`sel_func_name`)

For table views, specify the function for handling select statements in the transaction manager for the specified server/table view. The Select Handling property must be set to Function Name.

Select Handling (`select_handling`)

For table views, select the method for handling select statements in the transaction manager for the specified server/table view: SQL Statement Generation (`PV_HANDLING_SQL`), Function Call (`PV_HANDLING_FUNC`), or Nothing (`PV_HANDLING_NOTHING`).

Service Transaction (`tm_transaction`)

A runtime-only application property in JetNet and Tuxedo executables that determines whether a service is transaction-manager enabled and, if so, which transaction manager operation is to be performed.

Sort Order (`sort_order`)

Under Format/Display, specify the sort order to be used when the widget is in an array or in a grid. If the widget is in a grid, the Column Click Action property must also be set to Sort.

Sort Order Function (`sort_order_func`)

Under Format/Display, specify the custom function to be invoked when Sort Order is set to Custom. The function can be either a JPL procedure or prototyped C function.

Stylesheet Data (`stylesheet_data`)

Under Web Options, for inline style sheets, enter the style sheet specification.

Stylesheet Link (`stylesheet_link`)

Under Web Options, specify the URL location of the style sheet. On the HTTP server, the style sheet should be located in the public documents directory.

Stylesheet Source (`stylesheet_source`)

Under Web Options, specify whether the style sheet for the web application screen is included in the screen itself (Inline) or is in a separate document (Link).

Stylesheet Type (stylesheet_type)

Under Web Options, specify the type of style sheet to be used for the web application screen: CSS (cascading style sheets) or JavaScript.

Submit (submit)

Under Web Options, setting this new push button property to No will keep the screen from being submitted back to the web application server when the button is pressed.

Tab Entry Function (tab_entry_func)

For the index tab field on tab cards, under Focus, the name of the function to be called when the tab card is topmost and its index tab gains focus.

Tab Exit Function (tab_exit_func)

For the index tab field on tab cards, under Focus, the name of the function to be called when the tab card is topmost and its index tab loses focus.

Threshold (count_threshold)

For table view widgets, this property is a subproperty of the Count Warning property when Count Select and Count Warning are set to Yes. Use to specify the maximum number of rows to fetch in a result set. If a result set (stored in the server view's count_result property) exceeds this value, the user is prompted before the data is actually fetched.

Toolbar Allowed Sites (toolbar_allowed_sites)

For toolbars in Windows applications, a runtime application property sets the frame placement for the toolbar using one or more of the following bit flags: PV_TOOLBAR_FLOAT, PV_TOOLBAR_TOP, PV_TOOLBAR_BOTTOM, PV_TOOLBAR_LEFT or PV_TOOLBAR_RIGHT.

Toolbar Coordinates (toolbar_x_position, toolbar_y_position)

For toolbars in Windows applications, runtime application properties set the screen coordinates of the upper-left corner of the floating toolbar.

Toolbar Current Site (toolbar_current_site)

For toolbars in Windows applications, a runtime application property sets the current placement of the toolbar using one of the defined bit flags: PV_TOOLBAR_FLOAT, PV_TOOLBAR_TOP (default), PV_TOOLBAR_BOTTOM, PV_TOOLBAR_LEFT, or PV_TOOLBAR_RIGHT.

Toolbar Hidden (toolbar_hidden)

For toolbars in Windows applications, a runtime application property sets whether the toolbar is currently displayed using PV_YES and PV_NO. Users can hide the toolbar by clicking on the X in the upper-right corner of the menu.

Topmost Card (topmost_card)

For tab cards, a runtime-only property that specifies the card number of the card that is the topmost card in the tab deck.

Update Function Name (upd_func_name)

For table views, specify the function for handling update statements in the transaction manager for the specified table view. The Update Handling property must be set to Function Name.

Update Handling (update_handling)

For table views, select the method for handling update statements in the transaction manager for the specified table view: SQL Statement Generation (PV_HANDLING_SQL), Function Call (PV_HANDLING_FUNC), or Nothing (PV_HANDLING_NOTHING).

Warning (count_warning)

For table view widgets, this property is a subproperty of the Count Select property when Count Select is set to Yes. Use to specify whether the user is prompted, before the data is actually fetched, when the size of a result set (stored in the server view's count_result property) exceeds the value in the Count Threshold property.

Web ID (webid)

For Web applications, this application property obtains the name of the next cache file to be generated.

Property Changes

Fetch Directions/Directions (fetch_directions)

The table view Fetch Directions property has been renamed to Directions and is located in the new Server View category.

Both the table view Directions property and the screen Fetch Directions property have an additional value, none, which when set eliminates the possibility of doing CONTINUE command processing on a server view. CONTINUE functionality can consume system resources, therefore, using this property value can allow you to better control how a SELECT is issued against the table view.

Font properties

Screen and widget font properties that identify, what was JAM-specific fonts, have been updated to be Prolifics-specific fonts both in the Properties window and in the configuration map file; the JAM modifier has been eliminated.

Help properties (`help_screen`)

The menu property, `mni_jam_help` (menu item Help property) is now `mni_help`.

The screen property (JAM Help property) and its corresponding mnemonic, `jam_help_screen` are now Help Screen and `help_screen`, respectively.

Label (`label`) **property**

The Label property (`label`) is now available for grid frames. The setting provides a caption for the HTML table in Web applications.

Link (`default_link`) **property**

The Link property (`link`) in previous releases has been changed to Default Link (`default_link`).

In addition, business graphs can be assigned a URL. If no value is set, the graph does not act as an HTML link.

Memo Text (`memo1 . . . memo9`) **properties for table views and links**

Under Identity, both table view and link widgets can now have Memo Text properties assignments.

Relations (`relations`)

This property, which describes the relationship between two table views, has been refined into three sub-properties: `rel_child` (database column in child table view), `rel_parent` (database column in parent table view), and `rel_op` (type of relationship—join or lookup).

Style property (`style`)

The screen subproperty of the Pixmap property now defaults to Tile instead of Center. This only effects newly created screens.

Property Window Changes

Besides the new properties, some of the property categories were reorganized.

For text widgets, the properties listed under Database have been reorganized under new subheadings:

- Fetch Data includes Select-related properties (such as Use In Select and Use In Where).
- New Data includes Insert-related properties (such as Use In Insert).

- Change Data includes Update-related properties (such as Use In Update).
- Remove Data includes the In Delete Where property.

For table views, some of the properties previously listed under Database and Transaction have been moved to a new category: Server View.

Application Property Syntax

`@app()` replaces `@jam` as the property shortcut for the application name. `@jam` will continue to work for backward compatibility.

Readable Transaction Properties

All transaction manager properties are now readable at runtime via the property API. They include widget properties (under Column Edits): Length (`column_length`), Precision (`column_precision`), Scale (`column_scale`), and Type (`column_type`).

Writable Transaction Manager Properties

If a transaction manager transaction is not in effect, all transaction manager properties are writable.

JPL Programming

Declaring Variables

Use commas to delimit initial values in `GLOBAL` and `VARS` declarations.

Sending and Receiving Data

The `send` and `receive` commands have changed for word-wrapped fields. Word-wrapped fields are now sent as a single item; the `receive` command should specify a word-wrapped field which permits it to use `sm_ww_write` to place the text.

Variable Assignments

In previous versions of the product, an expression which mixed numeric with string variable assignments yielded inconsistent results. It is illegal to mix these assignments within one expression. For example, the following assignment previously yielded either 0 or an empty string, depending on which version is being used:

```
% .0 a='' // Assigned '' to a
```

Now, this assignment generates a syntax error.

Forwarding Messages

JetNet executables now contain support for `service_forward`, the JPL command that forwards service request data to another service.

Component-based Applications

Applications using COM components or EJBs have additional JPL commands:

- `log` — Writes a message to the `server.log` file.
- `receive_args` — Receives a method's parameters from a client.
- `return_args` — Returns a method's parameters back to the client.
- `raise_exception` — Sends an error code back to the client.

Application Property Syntax

`@app()` replaces `@jam` as the application property shortcut for the application name. `@jam` will continue to work for backward compatibility.

Library Functions

New Functions

`dm_convert_empty`

Determine if empty numeric fields should be replaced with a 0. This setting is database-specific since some databases do not allow NULL values in numeric columns.

dm_cursor_connection

Return the database connection for the specified cursor.

dm_cursor_consistent

Determine if the specified cursor is on the default connection.

dm_cursor_engine

Return the database engine for the specified cursor.

dm_disable_styles

Suppress the enforcement of styles in the transaction manager.

dm_enable_styles

Enable enforcement of styles in the transaction manager.

dm_get_db_conn_handle

Return a handle to the database connection's structure.

dm_get_db_cursor_handle

Return a handle to the database cursor's structure.

dm_get_driver_option

Return the value of a database driver option.

dm_odbc_preserves_cursor

Check to see whether the ODBC datasource preserves the cursor on a commit or a rollback.

dm_set_driver_option

Set the value of a database driver option.

dm_set_max_fetches

Set the maximum number of rows in a select set.

dm_set_max_rows_per_fetch

Set the maximum number of rows per fetch.

dm_set_tm_clear_fast

Clear all fields in a server view.

sm_com_load_picture

Get the object ID for the specified picture.

sm_com_QueryInterface

Access the `QueryInterface` method for the specified COM component.

sm_com_result

Get the error code returned by the last call to a COM component.

sm_com_result_msg

Get the error message returned by the last call to a COM component.

sm_com_set_handler

Set an event handler for the specified event on a COM component.

sm_get_tv_bi_data

Get before-image data.

sm_l_open_syslib

Opens a library as a system library.

sm_ldb_fld_get

Copy data from LDBs to specific fields.

sm_ldb_fld_store

Copy data from specific fields to LDBs.

sm_load_screen

Preload a screen into memory.

sm_log

Write a message to a server log from service components.

sm_menu_change

Set a menu's properties.

sm_mnitem_create

Insert a new item into a menu.

sm_msg_del

Delete a message set from memory.

sm_msg_read

Read messages from a memory block.

sm_mts_CreateInstance

sm_mts_CreateProperty

sm_mts_CreatePropertyGroup

sm_mts_DisableCommit

sm_mts_EnableCommit

sm_mts_GetPropertyValue

sm_mts_IsCallerInRole

sm_mts_IsInTransaction

sm_mts_IsSecurityEnabled

sm_mts_PutPropertyValue

sm_mts_SetAbort

sm_mts_SetComplete

For MTS executables, a set of wrapper functions to the associated COM/MTS method.

sm_mw_PrintScreen

In Windows executables, print Prolifics screens, sending either the current Prolifics screen or all the screens in the MDI frame to the printer.

sm_obj_call

Call a service component's method.

sm_obj_create

Instantiate a service component.

sm_obj_delete_id

Remove a service component.

sm_obj_get_property

Get the value of a property for a service component or ActiveX control.

sm_obj_onerror

Install an error handler for a service component.

sm_obj_set_property

Set the value of a property for a service component or ActiveX control.

sm_obj_sort

Sort the object's occurrences according to the rules specified in the object's Sort Order property.

sm_obj_sort_auto

Sort the object's occurrences according to the conventions for grids in Windows.

sm_raise_exception

Send an error code back to the client.

sm_receive_args

Receive the method's parameters from the client.

sm_return_args

Return a list of parameters back to the client.

sm_tm_handling

Process the specified transaction manager functions for special insert, update, select and delete handling.

sm_tm_old_bi_context

Specify the method of before-image processing.

sm_tp_get_svc_alias

For JetNet and Tuxedo applications, return the value of the service alias for the application server.

sm_unload_screen

Unload a screen from memory.

sm_validate

Validate a widget or the group of widgets in a container, such as a screen, tab card, or other group.

Changed Functions

dm_gen_change_select_list

For this function, do not use a local JPL variable as the target of a transaction manager fetch.

sm_inquire

A new parameter, `I_INERROR`, is available to determine if a message box is being displayed.

sm_tm_inquire, sm_tm_iset

A new argument, `TM_SV_SEL_COUNT` determines if an initial query will be performed in order to determine the number of rows in the select set.

Discontinued Functions

sm_ax Functions

The functions for components supersede the functions for ActiveX controls released in Prolifics 2.5.

ActiveX Function	Panther 4.2 Replacement
<code>sm_ax_call_method</code>	<code>sm_obj_call</code>
<code>sm_ax_get_prop</code>	<code>sm_obj_get_property</code>
<code>sm_ax_QueryInterface</code>	<code>sm_com_QueryInterface</code>
<code>sm_ax_result</code>	<code>sm_com_result</code>
<code>sm_ax_set_handler</code>	<code>sm_com_set_handler</code>
<code>sm_ax_set_prop</code>	<code>sm_obj_set_property</code>

sm_com Functions

The functions for components supersede the functions for COM components released in Panther 4.0 and 4.1.

COM Function	Panther 4.2 Replacement
<code>sm_com_call_method</code>	<code>sm_obj_call</code>
<code>sm_com_get_prop</code>	<code>sm_obj_get_property</code>
<code>sm_com_log</code>	<code>sm_log</code>
<code>sm_com_obj_create</code>	<code>sm_obj_create</code>
<code>sm_com_obj_destroy</code>	<code>sm_obj_delete_id</code>
<code>sm_com_onerror</code>	<code>sm_obj_onerror</code>
<code>sm_com_raise_exception</code>	<code>sm_raise_exception</code>
<code>sm_com_receive_args</code>	<code>sm_receive_args</code>
<code>sm_com_return_args</code>	<code>sm_return_args</code>
<code>sm_com_set_prop</code>	<code>sm_obj_set_property</code>

sm_fi_open

Is no longer documented. `sm_fi_open` was used to find a file (along the Prolifics's search path and open it in binary read-only mode. Use `sm_fi_path` instead to search along Prolifics's search path. Then call `fopen` (a standard C function) to open the file in any way you choose (it does not limit you to binary read-only mode).

sm_msgread

Has been replaced with the following new functions:

- `sm_n_msg_read` — Reads messages from a named file with standard file lookup protocol.
- `sm_d_msg_read` — Reads messages from the default message file (`SMMSGS` variable).
- `sm_msg_read` — Reads messages from a memory block.
- `sm_msg_del` — Deletes a message set from memory.

The message classes have also been updated; `FM_MSGS`, `JM_MSGS` and `JX_MSGS` messages are now located in `SM_MSGS`. The value for `WB_MSGS` has also been

updated. Any instances of `sm_msgread` in a Panther application should be updated to the new message classes.

Additional Flags for Widget Functions

Two additional flags are now documented for widget functions:

`K_EXTEND`

The widget is an extended selection list box.

`K_EXTEND_LAST`

For extended selection list boxes, the widget is the last item in the list box.

Development Environment

Java Interface

In addition to C and JPL, you can program your application behavior in Java. In the editor, Panther objects (screens, service components, widgets) can be assigned a Java tag, which defines a Java class to act as an event handler for that object.

At runtime, when a given object has an event handler associated with it, Prolifics will invoke the methods supported by the event handler in response to application events.

The event handler classes must provide methods that correspond to the various kinds of events supported by the object with which it is associated. To this end, predefined interfaces, that the event handler classes must implement, have been provided.

For more information on Java programming in Prolifics, refer to page 215 in the *Application Development Guide*.

Team Development

In JetNet and Tuxedo executables, developers can have personal copies of screens in library files and services in the JIF in order to make and test changes during development.

QA Partner Interface

In this release, you can call Prolifics functions from within QA Partner®. QA Partner, part of the QualityWorks™ software testing program manufactured by Segue Software Incorporated, can be used to automate functional and regression testing in client/server applications. For more information, refer to page 717 in the *Application Development Guide*.

Internal File Locking Available on Windows NT

Internal (native) file locking is now the default for Windows NT, and you will need to run `formlib -e` on a library in order to continue using external lock files. Once a library is set to external file locking with `formlib -e`, you must run `formlib -i` on that library in order to use internal file locking.

COM Components in Axview

In the Windows development environment, all COM components, not just ActiveX controls, are now displayed in `axview`.

Opening Library Files in Windows

In Windows executables, double clicking on the name of any file in a library will open the file in the program associated with it according to the Windows File Type setting.

MSVC Project Files

MSVC project files are now available for rebuilding your Prolifics executables.

Text Selection Keys

A new series of logical keys have been added for selecting text:

EXTFB	extend selection to start of field or list box
EXTFE	extend selection to end of field or list box
EXTL	extend selection with left arrow in text field
EXTLB	extend selection to start of line in text field
EXTLE	extend selection to end of line in text field
EXTPD	extend selection down one page in text field or list box
EXTPU	extend selection up one page in text field or list box
EXTR	extend selection with right arrow in text field
EXTWL	extend selection one word left in text field
EXTWR	extend selection one word right in text field
SLALL	select entire text field
SLWRD	select current word

In addition, EXTD and EXTU now also apply to text fields as well as list boxes.

Three-Tier Application Development

For more information about changes in JetNet and Tuxedo applications and about the open middleware connectivity functions for COM components and Enterprise JavaBeans, refer to page 43.

Configuration

Name extensions

Screen names no longer have a default value set for the application variable `SMFEXTENSION` which, in previous releases, specified the default file extension for screens. Refer to page 3 for more information.

Setting the JIF location

For JetNet and Tuxedo applications, you can now set the application variable `SMTPJIF` in your Windows initialization file.

IBM Visual Age for Java

For Panther/WebSphere applications, you can specify the command to launch IBM's Visual Age for Java program with `SMIBMVJAVA`.

IBM WebSphere Administrative Console

For Panther/WebSphere applications, you can specify the command to launch IBM's WebSphere Administrative Console program with `SMWSADMIN`.

Java support

The behavior variable `JAVA_USE` determines whether Java is initialized and the `Java not enabled` error message appears on startup. If `JAVA_USE` is set to `JAVA_NOT_USED`, the error message will never appear. If `JAVA_USE` is set to `JAVA_USED`, the error message will appear if Java has not been installed.

Java class factory

Specify your Java class factory with `SMJAVAFACTORY`. Otherwise, it defaults to `DefaultClassFactory`.

Java compilation

(optional) Specify the command used to compile Java with `SMJAVACOMPILE`. If unspecified, it defaults to:

```
javac -deprecation "%s"
```

On Windows, the default value is overridden by a setting in `smvars.bin`:

```
cmd /c javac "%s" || pause
```

Java editor

Specify your Java editor with `SMJAVAEDITOR`. Otherwise, it defaults to the value specified for `SMEDITOR`.

Java libraries

(optional) Specify the location of your Java libraries with `SMJVALIBRARY`. If set in the environment, it overrides the default location.

Motif resources

A new Motif resource has been added to the Prolifics resource file, `Prolifics*positionIsFrame`. When placing a window at a specific position on the display, the requested position can be for the placement of the frame or for the placement of the client window inside the frame. If the position is for the frame, set this resource to true (the default setting). The window manager can have a resource of the same name. The value of the Prolifics resource should match the value of the window manager. The distributed resource files are in the `config` directory.

Tuxedo support in Panther/WebSphere

In Panther/WebSphere applications, the initialization file (`panther.ini`) can set `SMTCLIENT` to specify whether Tuxedo connectivity is enabled and what type of

client is needed (native or workstation). Set `SMTPINIT` to specify the default arguments to the `client_init` command.

WebSphere Application Server

In Panther/WebSphere applications, specify URL of the machine running WebSphere Application Server in `SMPROVIDERURL`.

Utilities

New Utilities

Jam to Prolifics

A utility to help you upgrade your JAM application to Prolifics by packaging the application files into libraries. (Refer to page 59 in the *Upgrade Guide*.)

Web Setup Manager

A Web-based utility is available for creating and updating the files needed on your Web application server: the requester executable and the Web initialization file. Refer to page 125 in the *Web Development Guide* for a step-by-step guide.

COM/MTS Utilities

makedlls

In COM/MTS applications, a utility to generate the service component's DLLs for the specified libraries. (Refer to page 46 in *COM/MTS Guide*.)

WebSphere Utilities

makeejb

In Panther/WebSphere applications, a utility to generate the Java files for the service components in the specified libraries. (Refer to page 74 in *WebSphere Developer's Studio*.)

Changed Utilities

binherit

In addition to updating screens with inherited values from the repository, `binherit` also updates reports.

The following changes were implemented for the `-u` option:

- The `-u` option will be ignored for members of libraries that can only be opened read-only.
- If the `-u` option is not selected, libraries will be opened read-only.

f2asc

In addition to converting screens between binary and ASCII format, `f2asc` also converts reports and service components. ASCII files for screens and service components start with a `S`: output area containing screen/component properties; ASCII files for reports start with a `R`: output area. The output area for static labels has changed from `S`: to `L`:. The output area for the service component's interface starts with `I`:.

formlib

The new `-m` option compacts the library by removing unused space. Using this option before making the library read-only will allow the read-only operation to be reversible.

dd5upg

(JAM 5 updates only) The `dd5to6` utility has been renamed as `dd5upg`.

f5upg

(JAM 5 updates only) The `f5to6` utility has been renamed as `f5upg`. In addition, a new `-p` option includes the GUI interface values for the `hmargin`, `vmargin`, `hbuffer`, `vbuffer` properties in the converted screens.

monitor

In Panther, you must start the web application by using `monitor` or by installing the application as an Windows NT service which uses Services properties in the Control Panel to start the application.

`monitor` has a new option: `-restart` which combines the `clean`, `stop`, and `start` options. In addition, the syntax for `-install`, which installs the application as an NT service, has been expanded.

Discontinued Utilities

r2asc

The `r2asc` utility has been superseded by `f2asc`; therefore, to convert a report between binary and ASCII output, use `f2asc`.

rinherit

The `rinherit` utility has been superseded by `binherit`; therefore, to batch update all reports with inherited values from your application's repository, use `binherit`.

Database Interface

Improved SQL Processing

Currently, DBMS SQL statements that specify data modification and do not return data (`INSERT`, `UPDATE`, and `DELETE` statements) are executed by simply passing the SQL statement to the database immediately to process the statement quickly and efficiently. For SQL statements that return rows (`SELECT`), the process includes a prepare and execute cycle. This means that the database is first notified where to put the data (if any), and then tells the database to execute the SQL.

The method used to determine if a SQL statement returns rows is to execute the SQL statement and see if it returns rows. If it does, it goes through the prepare and execute cycle—essentially executing the `SELECT` statement twice. If the statement is a stored procedure which inserts a row and then selects back data, the stored procedure is executed twice and therefore causes two copies of the row to be inserted.

The new method of SQL statement processing includes two new DBMS statements which will improve `SELECT`-type processing and performance:

- `DBMS QUERY` — Executes the SQL statement based on the assumption that it may or may not return data.
- `DBMS RUN` — Executes the SQL statement immediately, which assumes that no data is returned from the database.

Performance is improved because:

- Data-modification (non-`SELECT`) statements are not executed twice.
- Since it can be determined ahead of time whether or not to expect fetched rows, it takes less time to execute a `SELECT` statement.

Specifying Variables in `DECLARE CONNECTION`

The new recommended syntax for `DBMS DECLARE CONNECTION` allows the values for the connection options to contain spaces or punctuation characters. Use

the WITH keyword in the statement (instead of FOR) and connect the option and value with an equal sign in comma-separated pairs. Variables no longer need to be colon-expanded; strings must still be in quotation marks. The following example contains two variables for the user and password and a quoted string for the database path:

```
DBMS DECLARE c1 CONNECTION WITH \
    USER=user, PASSWORD=pword, \
    DATABASE="C:\Program Files\Prolifics\videobiz"
```

Since the variables are not colon-expanded in this variant, the values will not appear in error messages and trace statements.

Importing Database Tables

The importer has been changed to allow you to specify whether the user name is included in the table name. On the Import Database Tables screen, choose Options to activate this feature.

JDB Databases

If you place your JDB database in a directory specified in SMPATH (or in your application directory), you no longer need to specify the full pathname in your DBMS DECLARE CONNECTION statement.

Database Drivers

The following changes were made to all database drivers:

BLANK_AS_NULL

A new driver option which converts all data bound as CHAR to be NULL if the source field is an empty string.

New dm variables

If the database can return SQL state information, that information is stored in @dmengsqlstate and @dmwarnsqlstate.

Sybase

In addition to the DBMS DECLARE CONNECTION statement, Sybase executables can also set logon options in usrhandlr.c.

Sybase using CT Library has two new connection options: SET_CURSOR_OPTIONS and SET_CURSOR_OPTIONS_DEFAULT.

ODBC

ODBC to SQLBase has a new connection option: `DISCONNECT_BEHAVIOR`.

Library Functions

The Library Functions section notes the addition of the following new functions:

- For obtaining information about the database connection handle:

`dm_get_db_conn_handle`

- For obtaining information about the database cursor:

`dm_cursor_connection`
`dm_cursor_engine`
`dm_cursor_consistent`
`dm_get_db_cursor_handle`

- For setting database driver connection options:

`dm_get_driver_option`
`dm_set_driver_option`

- For controlling how data is fetched:

`dm_set_max_fetches`
`dm_set_max_rows_per_fetch`

Transaction Manager

The Properties section notes the ability to update many traversal properties that were read-only in previous releases and the addition of the following new properties:

- `bi_string` — Access to before image values.
- `di_string` — Access to deleted rows.
- `num_columns` — The number of columns in the specified table view.
- `num_del_images` — The number of deleted rows.

The Library Functions section noted the addition of the following functions:

- For controlling how data is cleared in a server view:

`dm_set_tm_clear_fast`

- For controlling how styles are applied:

```
dm_disable_styles
dm_enable_styles
```

Transaction Manager Common Model

The previous set of database-specific transaction models delivered with Prolifics have been replaced with smaller, more manageable models. In addition to the database-specific model, each engine also accesses a common transaction model, containing the functionality common to all of the database engines. The source code for the new database-specific transaction models is provided and can be modified to make global changes in transaction manager functionality. The common model should not be modified; however, the source code is available for reference.

Having an database-specific model expands the event processing in the transaction manager. As in previous versions, the transaction manager first checks to see if an event function has been specified for the event. If so, it is processed; otherwise, the transaction manager proceeds to the database-specific transaction model. If database-specific processing for the event is required, it must be contained in this model. Otherwise, the transaction manager proceeds to the common transaction model and performs the processing defined there.

To call the common model in addition to an event function and the database-specific model, have the event processing in the event function and the database-specific model return `TM_PROCEED`, which passes the processing to the next level. The common model is always called for `TM_START` and `TM_FINISH` events.

The common model provides plausible processing for every event known to the transaction manager. This includes default behavior for the database transaction events. While a majority of the database-specific transaction models set the mode to initial after a database transaction is committed, the common model does not, leaving this for the database-specific transaction models.

If you have revised an existing transaction model, the revised version can continue to serve as a database-specific transaction model. Since none of the previously distributed transaction models return `TM_PROCEED`, unless this return value has been explicitly coded, the transaction manager will only access the common model for the new transaction manager events.

If you have implemented a transaction manager event function and use one of the existing transaction models, there is no visible effect with the replacement of the old models with the new.

New Transaction Manager Events

Some database-specific models have additional events for database transaction processing. The following slice events were added to the `SAVE` command:

- `TM_SAVE_BEGIN` — For some engines, start a database transaction with `DBMS BEGIN` or a savepoint with `DBMS SAVE`.
- `TM_SAVE_COMMIT` — For full `SAVE` commands, do a `DBMS COMMIT`. If it fails, push the `TM_SAVE_ROLLBACK` event onto the stack. If it succeeds, push the `TM_SAVE_SET_MODE` event onto the stack.
- `TM_SAVE_ROLLBACK` — Push the `TM_SAVE_SET_MODE` event onto the stack. Do a `DBMS ROLLBACK`.
- `TM_SAVE_SET_MODE` — Some engine-specific models set `TM_VALUE` to `TM_INITIAL_MODE` to indicate that processing should resume in initial mode. (These engines discard the select set when commits and rollbacks are performed.)

Releasing Database Cursors in the Transaction Manager

The transaction manager has a new command, `RELEASE`, which releases the database cursors when the transaction manager is active.

Calling Transaction Event Functions

The `WALK` commands direct the transaction manager to traverse the transaction tree of an application screen. These commands have no processing attached to them in the transaction models so the traversal can be used to fire any transaction event functions.

`WALK_SELECT` traverses the tree in select order, starting with the root table view and then by server view and table views within each server view. `WALK_INSERT` traverses the tree in insert order. `WALK_UPDATE` traverses the tree in update order. `WALK_DELETE` traverses the tree in delete order. The values of the link properties, Insert Order, Update Order, and Delete Order, can change the traversal order of the table views.

Checking the Size of the Select Set

If the Count Select (`count_select`) property is set to Yes, the transaction manager performs an initial SQL `SELECT` statement in order to determine the size of the select set and stores it in the `count_result` property. If the size is greater than the value of the Threshold (`count_threshold`) property and the Warning (`count_warning`) property is set to Yes, the transaction manager displays a message box, giving the user an opportunity to discontinue the data fetching operation.

To implement this processing, three new slices were added to the `TM_SELECT` and `TM_VIEW` request events: `TM_SET_SEL_COUNT_FLAG`, `TM_SEL_COUNT_CHECK` and `TM_CLEAR_SEL_COUNT_FLAG`. Also, `sm_tm_inquire` and `sm_tm_iset` have a new parameter, `TM_SV_SEL_COUNT`.

Support for Outer Joins

If a link widget's Type property is set to Server, the Join Type property allows you to specify the join operation of a `SELECT` statement that combines information from two database tables, if the database engine supports this feature. The Join Type property can be set to: Inner (`PV_INNER`) (default), Left Outer (`PV_LEFT_OUTER`), Right Outer (`PV_RIGHT_OUTER`), or Full Outer (`PV_FULL_OUTER`). For more information about how the Join Type setting affects the select set, refer to page 358 in the *Application Development Guide*.

Generating SQL Statements

On the Tools Menu, the Generate TM SQL option writes all of the transaction manager SQL statements to a file.

SQL Statement Handling

You can now specify whether the transaction manager should generate SQL for delete, insert, select, or update statements, call a separate function, or do nothing.

For table views, the Properties window contains a new category, Server View. All the properties for select statements have been moved to this category, under the Select Handling property. The properties for delete, insert and update statements are in the Database category, under Delete Handling, Insert Handling and Update Handling respectively.

Regenerating SQL

For insert and update statements, one of the new properties is Regenerate SQL. This property determines how the transaction manager generates SQL statements for these operations.

This property setting is most useful in grids with multiple occurrences where the transaction manager examines each row in order to generate the SQL statement, instead of basing the number of columns in the SQL statement on the changes in the first row.

Primary Key Updates

With the `primary_key_update` property, you can specify whether the transaction manager should update primary key columns by generating a SQL `UPDATE`

statement or by generating a SQL `DELETE` statement followed by a SQL `INSERT` statement.

Writable TM Properties and Default Transaction Name

If a transaction manager transaction is not in effect, all transaction manager properties are writable. You can determine the current transaction name using the `default_tran` property.

Web Application Development

Web Setup Manager

A new web-based utility, the Web Setup Manager, will help write and configure the files needed for your web application. Refer to page 125 in the *Web Development Guide* for a step-by-step guide.

Web Initialization Files

New Initialization Settings

Web initialization files have the following new initialization variables:

- `NumServers` — The number of Jserver processes, or concurrent users, for an application. This setting replaces `MinServers` and `MaxServers`. Web applications from previous releases need to update their web initialization file to the new variable.
- `IdleServerTimeout` — The number of seconds that a Jserver process will wait for an incoming request before exiting.
- `EnableWebid` — Activates the caching process which uses the `webid` property to obtain the cache file so that it can be specified in the URL.
- `ListenQueueLength` — The length of the listen queue. This approximately represents the number of web requests that can be waiting for an available Jserver.
- `PadOptionMenus` — For option menus, pads the text with trailing and HTML spaces (`nbsp`) if set to Yes. To pad only the first occurrence, set this option to First. Setting this option to Yes matches the behavior in previous versions of Panther.

One Initialization File In previous releases, a Prolifics Web application read the values from `proweb.ini` file before reading your application's initialization file (`appName.ini`). This was done so that the `proweb.ini` could define any global values which are common among all Web applications, but it caused problems for application maintenance. Therefore, the distributed `proweb.ini` file will no longer be read when you have an application initialization file. Only one initialization file is read for each Web application.

If you are using a global initialization file (`proweb.ini` or `jamweb.ini`) to set global parameters, merge all global values into your application-specific initialization file.

Caching Application State

The state of the application can now be obtained when performing a `GET` for Prolifics files. In previous releases, invoking screens and reports via a `GET` caused the state information to be lost. The following can now be accomplished:

- Hyperlinks — A hyperlink can be used to obtain a Prolifics screen that has access to application state information on the server.
- Frames — The HTML file which defines the frames would be set as an HTML template. Then, using the procedure for HTML templates, the `<FRAMESET>` can specify a series of screens sharing the same cache file.
- HTML Template — The name of the cache file can be encoded into the HTML template using the `<<widget_name>>` syntax. The template can call subsequent Prolifics screens via `GET` with this cache name specified. The called screens would then have access to the cache information.

To implement this caching behavior, two new application properties are available:

Previous Form (`previous_form`)

Gets the screen name as stored in the current cache file. Typically, this would be the name of the last screen that was accessed.

WebID (`webid`)

Obtains the name of the next cache file to be generated.

To access the cache file, a new `name=value` pair can be encoded as part of the URL:

```
@webid=cacheFile
```

For more information, refer to page 53 in the *Web Development Guide*.

Web Entry Processing

When screens are submitted at runtime, Prolifics variables (`@web_action`, `@web_action_widget` and `@web_action_occurrence`) contain information about the push button that was pressed and the widget's object ID and occurrence number, if applicable. These variables can be accessed in `web_enter` processing. For more information, refer to page 49 in the *Web Development Guide*.

HTML Template Processing

You can now include conditional processing in HTML templates with the following constructs:

```
{{include:filename}}
{{while:condition}}
{{if:condition}}
{{else:}}
{{elseif:condition}}
{{end:}}
{{eval:statement}}
```

Two application properties are associated with HTML loop processing: `html_max_loop` to limit the number of loop iterations and `html_max_nest` to limit the number of nesting levels.

For more information, refer to page 68 in the *Web Development Guide*.

Web-specific Properties

The Properties section also listed the following new Web-specific properties and property changes.

Grid Frames

There are new properties for a grid frame widget implemented for Web applications that let you control the creation of the default controls appearing in the HTML representation of the grid frame. The new Web Option properties are:

Scroll Buttons property (`scroll_buttons`)

If set to Yes (default), allows scroll buttons (Page Up, Page Down, Top, and Bottom) to appear on the grid when the number of occurrences in the grid exceed the number of onscreen rows. If set to No, the scroll buttons are not generated under any circumstances.

Insert/Delete Buttons property (`ins_del_buttons`)

If set to Yes (default), Insert (Insert Above and Insert Below) and Delete buttons are generated for the grid frame. If set to No, the buttons are not generated under any circumstances.

	<p>Radio Buttons (<code>radio_buttons</code>)</p> <p>If set to Yes (default), radio buttons are generated for each occurrence in the grid frame if one of the columns is unprotected or if the grid's Stripe Current Row property is set to Yes.</p> <p>In the absence of Prolifics buttons, you can create custom controls to perform the same functionality, but have better control of the appearance and functionality of buttons used to navigate and manipulate grids on the Web.</p>
Graphs as URL Link	<p>You can assign a URL to a graph widget so that a user can click on the image and proceed to the specified URL.</p> <p>Link property (<code>default_link</code>)</p> <p>Under Web Options, this property lets you assign a URL to the graph. If no value is set, the graph does not act as an HTML link.</p>
New Browser Event	<p>For JavaScript and VBScript, the OnMouseOut event for widgets is now available.</p> <p>OnMouseOut property (<code>on_mouse_out</code>)</p> <p>Under Browser Events, this property lets you specify a JavaScript or VBScript function to execute when the mouse pointer leaves an area (in client-side image maps) or a link.</p>
Running Client-Side Scripts	<p>Instead of submitting the screen back to the server, clicking on a push button can run JavaScript or VBScript on the browser.</p> <p>Submit property (<code>submit</code>)</p> <p>Under Web Options, setting this property to No will keep the button from causing a Submit action, allowing the button to be used for browser-side actions defined in JavaScript or VBScript.</p>
Setting a Wallpaper	<p>The default value of the Style property has changed to allow a wallpaper effect for Web application screens.</p> <p>Style property (<code>style</code>)</p> <p>The screen subproperty of the Pixmap property now defaults to Tile instead of Center. This only effects newly created screens.</p>
Specifying Style Sheets	<p>Style sheets can be specified for web application screens. The style sheet can be included in the web screen itself or in a separate style sheet file.</p> <p>Stylesheet Data (<code>stylesheet_data</code>)</p> <p>Under Web Options, for inline style sheets, enter the style sheet specification.</p>

Stylesheet Link (`stylesheet_link`)

Under Web Options, specify the URL location of the style sheet. On the HTTP server, the style sheet should be located in the public documents directory.

Stylesheet Source (`stylesheet_source`)

Under Web Options, specify whether the style sheet for the web application screen is included in the screen itself (Inline) or in a separate document (Link).

Stylesheet Type (`stylesheet_type`)

Under Web Options, specify the type of style sheet to be used for the web application screen: CSS (cascading style sheets) or JavaScript.

Hyperlinks

There are new properties for hyperlinks in Web applications:

Default Link (`default_link`)

Specify the URL location for this hyperlink. (This replaces the `link` property in previous releases.) If this property is specified for an array, it specifies the hyperlink location for every occurrence in the array.

Item Link (`item_link`)

Specify the URL location for this hyperlink. If specified for a particular element in an array, this property overrides the value in the Default Link and Image Map properties. Using this property, you can specify a different URL location for each occurrence.

Determining Mouse Location

Two JPL globals, `@web_image_click_x` and `@web_image_click_y`, contain the X and Y coordinates of the user's mouse click for use in JPL procedures.

Requester Executables

If you are using an ISAPI- or NSAPI-compliant HTTP server, use the new ISAPI and NSAPI versions of the requester executable, instead of the CGI version, for faster processing of your HTTP requests.

Java Servlets

A Panther web application can run as a Java servlet. For more information, refer to page 139 in the *Web Development Guide*.

Windows NT Service

The `monitor` command has a new syntax and contains additional parameters to use when installing your Prolifics Web application as an NT service.

Widget Positioning in Web Applications

In order to control widget positioning in Web applications better, a `COLS` attribute has been added to the table definition in the generated HTML. This will affect the widget positioning for screens built in previous versions of Prolifics. For new screens, there is a higher correlation between the GUI position and the HTML position.

Web Wizard Defaults

For wizard-generated Web screens, the default values have changed for some properties. The `Border`, `Title Bar`, and `System Menu` properties now default to `No`.

Naming Conventions

With the change in naming conventions, `smrepost.jam` becomes `smrepost.scr`.

Reports

Report Utilities

The `r2asc` utility has been superseded by `f2asc`; therefore, to convert a report between binary and ASCII output, use `f2asc`. The `rinherit` utility has been superseded by `bininherit`; therefore, to batch update all reports with inherited values from your application's repository, use `bininherit`.

Three-tier Development

In addition to JetNet and Tuxedo:

- COM components can be deployed under COM or MTS.

With the expanded middleware architecture, the term *request broker* was changed. In its place, the documentation uses the terms *middleware*, *middleware API*, and *middleware session*.

The term *service container* was changed to *service component*.

JetNet and Tuxedo

Creating service components

There are menu options for creating, opening and saving service components.

Forwarding service requests

`service_forward`, the JPL command that forwards service request data to another service, is now available for the JetNet applications.

Environment files for application servers

The environment files for the application servers have changed. A new file, `machine.env`, specifies the settings for the machine. `devserv.env` will use this new environment file, and no longer need a separate environment file. Some of the settings in `proserv.env` have changed.

JetMan now displays the location of the environment file and enters a default value, allowing you to update the setting if needed.

File access servers

The development access server is now called the file access server.

Workstation listener

In JetMan, you can specify the Handler Port property in the Machine Networking screen to set a port range for the workstation listener process.

`tlisten.pw` support

`rblisten` now uses the `tlisten.pw` file to authenticate most service requests by checking that any process requesting a service contains at least one of the passwords found in the file. If a file named `.adm/tlisten.pw` in the application directory is not found, the passwords are obtained from the file `$SMBASE/uda-taobj/tlisten.pw`.

Service aliasing

During development, you can now test services under a service alias instead of modifying the original service. This allows developers to continue to use existing services while service changes are being tested.

To use this feature:

- The application server must have a service alias defined. Once defined, the library function `sm_tp_get_svc_alias` returns the value of the service alias for the application server.
- An existing JIF file from a previous release of Prolifics must be opened and saved in order to enable service aliasing.
- The user identifier must be defined in Options⇒Service Alias.

For more information, refer to page 67 in the *Application Development Guide*.

Team development

In JetNet and Tuxedo executables, developers can have personal copies of screens in library files and services in the JIF in order to make and test changes during development.

Open Middleware Connectivity

If you specify the type of component system currently in use, you can use the same C functions to call methods and get/set properties on either COM components or Enterprise JavaBeans.

- Use the `current_component_system` property to specify the type of components currently in use: `PV_SERVER_COM` for COM components or `PV_SERVER_EJB` for Enterprise JavaBeans deployed under WebSphere Application Server.
- After specifying the `current_component_system` property, instantiate a component with `sm_obj_create`.
- `sm_obj_delete_id` destroys the component.
- `sm_obj_call` calls a component's method.
- `sm_obj_get_property` and `sm_obj_set_property` get and set a component's properties.
- `sm_receive_args` gets the method's in and in/out parameters; `sm_return_args` returns the out and in/out parameters back to the client.
- `sm_log` writes an error to a server log.
- `sm_raise_exception` sends an error code back to the client.

- `sm_obj_onerror` installs an error handler.

Panther WebSphere Applications

For information on building and deploying Enterprise JavaBeans under WebSphere Application Server, refer to page 1 in *WebSphere Developer's Studio*.

Documentation

Documentation titles

The titles of some manuals have changed in this release:

Prolifics 2.5 Title	Panther Title
<i>Administration Guide</i>	Online under <i>Enterprise Administration: JetNet Guide/Tuxedo Guide COM/MTS Guide WebSphere Developer's Studio</i>
<i>Application Development Guide</i>	<i>Application Development Guide</i>
<i>Editors Guide</i>	<i>Using the Editors</i>
<i>Language Reference</i>	<i>Programming Guide</i>
<i>Tutorial</i>	<i>Getting Started</i>
<i>Update Guide</i>	<i>New Feature</i>

A new *Quick Reference* manual has been printed, containing a list of the property names, library functions, JPL commands, and transaction manager commands. The properties reference section of that manual is available online.

The *Application Development Guide* has been totally re-organized in order to illustrate a typical development process.

Online documentation changes

Panther documentation is available in PDF format for the Panther 4.25 documentation set and in HTML for two books: *Panther for IBM WebSphere Installation* and *WebSphere Developer's Studio*. For more information about Panther online documentation, refer to Appendix A in *Panther for IBM WebSphere Installation*.

Online Documentation Changes and Corrections

Other documentation changes not listed in previous sections are:

ActiveX controls

Refer to page 883 in the *Programming Guide* for documentation of the ActiveX controls that are distributed with Prolifics.

ASCII JPL modules

Even though it is recommended that all JPL modules be placed in libraries, JPL modules can be in ASCII format on disk. This has been added back to the Programming in JPL chapter.

CGI variables

In Web applications, the variables containing the HTTP header information, such as `@cgi_request_method`, are now referred to as HTTP server variables.

cgi-bin directory

With the addition of ISAPI and NSAPI requester executables, the `cgi-bin` directory is now referred to as the program directory or scripts directory.

Java

Panther 4.0 and 4.1 documentation incorrectly included a row parameter for the `gridRowEntry` and `gridRowExit` events.

Reports

Panther 4.0 documentation incorrectly included descriptions of the Bar Height and Portable Placement properties. These properties are not in the Panther product.

sm_card_val

sm_tw_val

The first release of Panther documentation incorrectly included these functions. For tab card validation, use `sm_validate`.

sm_msg_del

sm_msg_read

Initial releases of the Panther documentation did not have an updated list of the message prefixes or message classes. `FM_MSGS`, `JM_MSGS` and `JX_MSGS` messages are now located in `SM_MSGS`. The value for `WB_MSGS` has also been updated.

sm_prop_get_str

The documentation has been updated to say that this function stores the returned data in a pool of buffers that it shares with other functions so you need to copy or process this data immediately.

Traversal properties for table views

Traversal properties for table views and server views in the transaction manager return the table view or server view name as a string, not as an object id.

Quick Reference Changes and Corrections

The Panther Quick Reference which was printed in November 1999 does not have the following changes:

Configuration

Text Selection Keys – Windows

Refer to page 26 for the list of new logical keys.

SMIBMVJAVA, SMPROVIDERURL, SMTPCLIENT, SMTPINIT, SMWSADMIN

Refer to page 26 for a description of the configuration variables in Panther for IBM WebSphere.

Functions

sm_com Functions

The functions for components supersede the functions for COM components released in Panther 4.0 and 4.1.

COM Function	Panther 4.2 Replacement
sm_com_call_method	sm_obj_call
sm_com_get_prop	sm_obj_get_property
sm_com_log	sm_log
sm_com_obj_create	sm_obj_create
sm_com_obj_destroy	sm_obj_delete_id
sm_com_onerror	sm_obj_onerror
sm_com_raise_exception	sm_raise_exception
sm_com_receive_args	sm_receive_args
sm_com_return_args	sm_return_args
sm_com_set_prop	sm_obj_set_property

Properties

sm_obj_sort

sm_obj_sort_auto

New functions for sorting data in arrays and grids.

Column Click Action (`column_click_action`)

For widgets in grids, under Format/Display, specify the action—sort or custom function—that occurs when a user clicks on the grid column header.

Column Click Function (`column_click_func`)

For widgets in grids, under Format/Display, specify the custom function to invoke when a user clicks on the grid column header. For this property to be available, Column Click Action must be set to Custom.

Connection Pooling (`conn_pool_size`)

In Panther/WebSphere applications, specify the number of concurrent database connections.

Current Component System (`current_component_system`)

A runtime-only property that instantiates the type of component system currently in use. Before creating any service components, set this property to `PV_SERVER_COM` for COM components or `PV_SERVER_EJB` for Enterprise JavaBeans deployed under WebSphere Application Server.

HTML Max Loop (`html_max_loop`)

For HTML templates using condition processing, specify the number of loop iterations to perform before terminating the process. The default setting is 1000.

HTML Max Nest (`html_max_nest`)

For HTML templates using condition processing, specify the number of nesting levels. Each `if`, `while`, or `include` constitutes one level. The default setting is 20.

In Server (`in_server`)

A new value for Panther/WebSphere applications, `PV_SERVER_EJB`.

Max Bundles (`max_bundles`)

A runtime-only application property specifying the number of JPL bundles available for `send` and `receive` commands. It defaults to ten bundles (including the unnamed bundle) if unspecified.

Provider URL (`provider_url`)

For Panther/WebSphere applications, a runtime-only application property specifying the location of the WebSphere application server machine. If

SMPROVIDERURL is set in the environment, the property is initially set to this value.

Runtime License (`runtime_license`)

For ActiveX controls which support runtime licensing, if the Runtime License property exists, the control will be created using the license.

Sort Order (`sort_order`)

Under Format/Display, specify the sort order to be used when the widget is in an array or in a grid. If the widget is in a grid, the Column Click Action property must also be set to Sort.

Sort Order Function (`sort_order_func`)

Under Format/Display, specify the custom function to be invoked when Sort Order is set to Custom. The function can be either a JPL procedure or prototyped C function.

Utilities

formlib

New `-m` option for compacting the library.

makeejb

Panther/WebSphere utility for generating the Java files for EJBs from the service components in an application library.



New in Panther 4.2

The documentation additions for Panther 4.2 are:

- ActiveX Runtime License property (`runtime_license`).
- Column Click Action property (`column_click_action`) and Column Click Function property (`column_click_func`).
- `formlib -m` option.
- Grid row margin settings with the Default Row Margin (`default_row_margin`) and Row Margin (`row_margin`) properties.
- Open middleware connectivity functions and JPL commands (page 43) allowing you to use the same commands and functions with different middlewares.
- Property for the maximum number of JPL bundles (`max_bundles`).
- Sorting data with the Sort Order property (`sort_order`), Sort Order Function property (`sort_order_func`), and `sm_obj_sort`, `sm_obj_sort_auto` functions.
- Text selection keys in Windows (page 26).
- Web applications as Java servlets (page 139).
- Web initialization files have new options: `ListenQueueLength` and `PadOptionMenus`.

- Web HTML template changes (page 38) and the associated properties, `html_max_loop` and `html_max_nest`.
- For `sm_com_*` functions, the functions for components supersede the functions for COM components released in Panther 4.0 and 4.1.

COM Function	Panther 4.2 Replacement
<code>sm_com_call_method</code>	<code>sm_obj_call</code>
<code>sm_com_get_prop</code>	<code>sm_obj_get_property</code>
<code>sm_com_log</code>	<code>sm_log</code>
<code>sm_com_obj_create</code>	<code>sm_obj_create</code>
<code>sm_com_obj_destroy</code>	<code>sm_obj_delete_id</code>
<code>sm_com_onerror</code>	<code>sm_obj_onerror</code>
<code>sm_com_raise_exception</code>	<code>sm_raise_exception</code>
<code>sm_com_receive_args</code>	<code>sm_receive_args</code>
<code>sm_com_return_args</code>	<code>sm_return_args</code>
<code>sm_com_set_prop</code>	<code>sm_obj_set_property</code>



New in Panther 4.25

The documentation additions for Panther 4.25 are:

- Enterprise JavaBean support is available in Panther for IBM WebSphere. Two new manuals were written, *Panther for IBM WebSphere Installation* and *WebSphere Developer's Studio*, as well as changes made throughout the Panther documentation set.
- Current component system property (`current_component_system`) has a new value: `PV_SERVER_EJB`.
- Server property (`in_server`) has a new value: `PV_SERVER_EJB`.
- Database connection pooling property (`conn_pool_size`) is available in Panther for IBM WebSphere to set the number of concurrent database connections.
- New application variables are available in Panther for IBM WebSphere:
 - `SMPROVIDERURL` — Specify the URL of the server machine running WebSphere Application Server. This value is available at runtime through the application property, `provider_url`.
 - `SMIBMVJAVA` — Specify the full pathname to launch IBM's Visual Age for Java program.
 - `SMTCLIENT` — Specify whether Tuxedo connectivity is enabled.

- `SMTPINIT` — Specify the default arguments to the `client_init` command.
- `SMWSADMIN` — Specify the full pathname to launch IBM's Administrative Console program.
- `makeejb` — Generates the EJB's Java files for the service components in the specified libraries.
- In the editor, the Component Interface window has an EJB tab where you specify settings for Enterprise JavaBeans.
- The Tools menu in the editor has two new options:
 - IBM VisualAge for Java — (WebSphere applications) Starts IBM's Visual Age for Java program.
 - IBM WebSphere Administrative Console — (WebSphere applications) Starts IBM's WebSphere Administrative Console where you install and deploy EJBs.



New in Panther 4.5

The additions for Panther 4.5 are:

- Frameset support with splitter widgets in Windows applications (described in *Using the Editors*). Splitter widgets divide a frameset window into subareas, allowing each subarea to be populated separately.
- Tooltip support in Windows applications (described in *Using the Editors*).
- Options for window display in Windows applications with the following properties:
 - Keep In Frame property—Specifies whether the window is in the MDI frame.
 - Parent Window property—Specifies whether the parent window is the MDI frame.
 - Keep On Top property—Specifies whether the window remains topmost.
- Status line functions for all platforms:
 - `sm_sb_delete`—Deletes a status bar section.
 - `sm_sb_format`—Formats a status bar section.
 - `sm_sb_gettext`—Gets contents of a status bar section.
 - `sm_sb_insert`—Inserts a status bar section.

- `sm_sb_settext`—Sets contents of a status bar section.



New in Panther 5

The additions for Panther 5 are:

- XML support is available allowing you to import and export data from your Panther screens in XML format. For this feature, there are new properties and new functions:
 - A new XML category in the screen editor contains the Tag, Attributes, Prefix and Postfix properties available for screens and most widgets.
 - `sm_xml_export`—Returns a character string containing the XML data.
 - `sm_xml_export_file`—Writes the XML data to the specified filename.
 - `sm_xml_import`—Returns a character string containing the XML data.
 - `sm_xml_import_file`—Reads the XML data from the specified filename.
- Email support is available in Panther screens. You can configure the email headers and send text emails with attachments. The global mail object `PR_MAIL_SYSTEM` accepts the information needed to communicate with MAPI (the Windows Messaging API). Using a series of properties and functions, you can specify mail options, including attachments. This feature is described in the *Application Development Guide*.
 - `sm_mail_new`—Creates a new mail message object.

- `sm_mail_send`—Sends the mail message.
 - `sm_*mail_attach`—Attaches the specified file.
 - `sm_*mail_widget`—Sends an image of a Panther widget in the mail message.
 - `sm_*mail_text`—Sends the contents of a Panther field in the mail message.
 - `sm_*mail_file_text`—Sends the contents of the specified file in the mail message.
 - `sm_mail_messagee`—Sends a basic mail message.
- Framesets with splitter widgets are now available in Motif applications. (They were already available in Windows applications.) Framesets are described in *Using the Editors*. Splitter widgets divide a frameset window into subareas, allowing each subarea to be populated separately.
 - Tab card and tab deck widgets are now available in Motif applications. (They were already available in Windows applications.) Tab decks and tab cards are described in *Using the Editors*.
 - Tooltips are now available in Motif applications. (They were already available in Windows applications.) Tooltips are described in *Using the Editors*.
 - In Panther for IBM WebSphere, support for WebSphere 5 has been added.