

# **ProGen Plus 7.0**

## **New Features & Upgrader's Guide**

**EXCLUSIVE DISTRIBUTION AND MARKETING RIGHTS  
BUSINESS COMPUTER DESIGN INTERNATIONAL, INC  
950 YORK RD  
HINSDALE, IL 60521**

**(c) 1986-2001 ESDI**

# ProGen Plus 7.0 New Features Guide

## Table Of Contents

<b>INSTALLATION AND UPGRADING</b> .....	<b>3</b>
<i>Overview and requirements</i> .....	3
OVERVIEW OF THE UPGRADE PROCESS .....	4
<i>Product Library Conversion</i> .....	4
<i>Copying and Converting Program Definitions and Environments</i> .....	5
TO UPGRADE TO THE LATEST RELEASE OF PROGEN PLUS 7.0X .....	5
POST INSTALLATION REVIEW .....	6
<i>Shell Source File Libraries</i> .....	7
<i>Reconciling Program Contexts</i> .....	7
<i>Reconciling Program Fields</i> .....	8
FINAL CLEANUP TASKS.....	8
ADDITIONAL ASSISTANCE.....	8
<b>INTRODUCTION: OVERVIEW OF NEW FEATURES</b> .....	<b>9</b>
KNOWLEDGE BASE .....	9
<i>Using the KB</i> .....	9
<i>KB Technical Details</i> .....	10
ACTIONS EDITOR/WORK FIELD FEATURES (DATE/TIME FUNCTIONALITY ILE ONLY) .....	11
<i>*DATTIM Work Fields</i> .....	11
<i>New Date Actions</i> .....	11
ADDITIONAL DESIGN OPTIONS.....	13
<i>Optionally Block Window Lookups on Output-only Fields</i> .....	13
<i>*PANEL Work Fields Support Initialization Value</i> .....	13
<i>Additional Validation on the Data and Time Data Type Functionality</i> .....	13
<i>EVAL Action Supports Operation Extenders</i> .....	13
<i>Enhanced RPDA/PDA mobility</i> .....	14
<i>ZWRKDATCVT supporting ILE interface</i> .....	14
<i>AUTOINC Variable</i> .....	14
<i>Optional Default PDA Text Placement</i> .....	14
<i>Default Designer ID Settings</i> .....	14
<i>Direct Command Line Access</i> .....	14
<i>Partial Panel Redesigning</i> .....	14
<i>Flat Panel Rebuilding Functionality Improvement</i> .....	14
<i>Reorganize F16 Alt view window</i> .....	15
<i>PDM Compliance in the ProGen Design Tool</i> .....	15
<i>PDM Compliance Issues for Generated Programs</i> .....	15
<i>Direct Work Field and Actions Access</i> .....	15
<i>*RESULT Work Field Calculation Enhancements</i> .....	15
<i>Read all Records Support</i> .....	15
<i>Non-keyed File Support in Style V</i> .....	15
<i>Enhanced Designer Validations</i> .....	15
<i>Improvements to the *TOTAL processing</i> .....	15
ADDITIONAL REPORT CONTROL FIELDS.....	16
ADDITIONAL RPDA ENHANCEMENTS.....	19
<i>Support Free format extended DDS keywords for external printer files</i> .....	19
<i>Supported DDS Keywords</i> .....	19
<i>*VALUE Work Fields Displaying Values While in the RPDA</i> .....	23
<i>Blank After Option on Reports</i> .....	23

# Installation and Upgrading

---

## Overview and requirements

This document is intended for existing users of ProGen Plus who are upgrading their software to a newer release level, or moving ProGen Plus from a CISC to a RISC machine. This document is applicable to these scenarios:

- Upgrading from ProGen Plus 6.0x to ProGen Plus 7.0x

**Note:** The “x” next to the product release level denotes any interim version of the release, such as 6.05, the latest interim release of ProGen Plus release 6.0.

It should be read by those responsible for loading ProGen Plus 7.0x and converting the software.

## Current ProGen Plus Release

The install program for upgrader’s is designed to upgrade a minimum Release of 6.01 or higher to Release 7.0x. If you are not currently at Release 6.01 or higher, please contact BCD for the appropriate instructions on how to upgrade to Release 7.0x.

## Operating System

The RISC version of ProGen Plus requires OS/400 V3R7M0 or higher.  
The CISC version of ProGen Plus requires OS/400 V3R2M0.

## What this document contains

This document contains the following information:

- Overview of the upgrade process
- How to Upgrade to Release 6.0x
- Post Installation Review

**PLEASE READ EVERYTHING IN THIS SECTION BEFORE PROCEEDING**

## Overview of the Upgrade Process

The upgrade process to Release 7.0x will be performed without impacting the contents of your existing 6.0x library. If for any reason the installation process fails, your existing ProGen Plus library will still be available and ready to use. Any objects you may have created and placed in GN#LIB (generated source, programs and validation modules) will be retained following installation.

### The following steps are performed:

- You restore the BCD product installation library BCDINSTALL to your system. Then execute the command BCDINSTALL/BCDINSTALL. You will then be prompted to restore the Release 7.0x installation library called GN#INST to your system. This library contains the necessary installation and conversion programs. It also contains the product library for ProGen Plus 7.0x.
- Your current release of ProGen Plus resides in a library called GN#LIB. Your existing ProGen Plus library will be renamed during the install. You will be prompted for the name of this library, the default is GN#LIBOLD. If you have upgraded before, GN#LIBOLD may already exist on your system. If so, delete this library or choose another name. In order to rename the library, no users can be using GN#LIB, and it must not exist in the library list of any active session or job. You can verify this using the WRKOBJLCK command.
- ProGen Plus Release 7.0x will be restored from a save file into library GN#LIB.
- If requested, the ProGen Plus demonstration library will be restored from a save file.
- The technical support library BCDSUPPT will be restored, if it does not already exist on your system.
- The conversion process converts your product library from 6.0x to version 7.0x.

---

## Product Library Conversion

All data areas are moved from GN#LIBOLD to GN#LIB

The following files are copied and converted if necessary.

- GN#USRC - User-calcs entered through via the USRCOD action
- GN#VSRC - Validation modules source
- GN#CSRC - Source members for equations
- GN#FHFF - Central repository: File descriptions
- GN#FDFF - Central repository: Field descriptions
- GN#DSNRF- Designer ID file
- GN#DSPRF- Designer ID style protection file
- GN#FPPF - Program fields file - copied and merged
- GN#CNTXF- Program contexts file - copied and merged
- GN#HTHF - Help text header file
- GN#HTDF - Help text detail file
- QRPGSRC - RPG source file
- QRPGLASC - RPGLE source file
- QDDSSRC - DDS source file
- QCLSRC - CL source file

The final conversion step is to copy all non-ProGen Plus objects from library GN#LIBOLD to GN#LIB. These are objects you may have placed in the Release 6.0x library (objects such as generated programs, display files, validation modules, generated source etc.).

---

## Copying and Converting Program Definitions and Environments

After the product library is converted the program definitions found in GN#LIBOLD are copied and converted to Release 7.0x format. Following the conversion of GN#LIBOLD, a search is done on your system for ProGen Plus 6.0x environment libraries. A list of these libraries is presented. Each 6.0x library that is actively used should be converted to 7.0x format. Environment libraries can be converted interactively or in batch. The GN#\* files in each environment are saved to a save file ENVLIBSAVF prior to conversion. This save file exists in the converted environment library and can be used for recovery purposes. See the "Post Installation Review" section for more information.

**Follow the information on the panels as the installation is progressing**

## To Upgrade to the latest Release of ProGen Plus 7.0x

### Install Steps

To upgrade to the latest release of ProGen Plus 7.0x from an earlier release of ProGen Plus, do the following:

1. Allow 1 to 2 hours to complete the install, depending on the relative performance of your AS/400, the size of your existing ProGen Plus library and the number of environment libraries.
2. Ensure that you have up to date system backup including GN#LIB and all ProGen Plus environment libraries.
3. Sign on as QSECOFR, or with a user profile whose group profile is one of these.
4. Make sure no-one on the system is using library GN#LIB. This library name must not exist in any session's library list. Run a WRKOBJLCK QSYS/GN#LIB \*LIB to ensure no locks exist.
5. Load the magnetic media containing the software on the appropriate device on your AS/400.
6. Ensure that the library GN#INST does not exist on your system. If it is found, delete it.

**BCD uses a generic install library called BCDINSTALL for all its products. If you have previously installed other software from BCD from the current tape or CD, check if the library BCDINSTALL still exists on your system. If it does, continue with step 11. If you have just received new media, delete the library BCDINSTALL (if it already exists on your system), and continue with step 7.**

7. If you are restoring the installation library from tape, type the following at a command line:  
**RSTLIB BCDINSTALL DEV(xxxxxxxx) VOL(\*MOUNTED)**  
where xxxxxxxxxxx is the device name.
8. If you are restoring the installation library from a CD, type:  
**RSTLIB SAVLIB(BCDINSTALL) DEV(OPT01) OPTFILE('/AS400LIB/BCDINST.LIB')**
9. When the library has been restored, type the following at an AS/400 command line:  
**BCDINSTALL/BCDINSTALL**
10. A panel will appear displaying a list of products available for installation. Type 1=Select next to ProGen Plus and press Enter. Next, you will be prompted for the device to install from. Enter the appropriate device name (the same as you specified in step 7 or 8) and press Enter.
11. The installation process will restore the library GN#INST, the ProGen Plus install library. Once restored, the installation process will automatically continue with the next steps, described below.
12. A panel appears with summary information about the install process. You will be prompted for a the name your existing product library (GN#LIB) should be renamed to and if you want to install the demonstration applications. Type Y=Yes to restore the demonstration library called GN#DEMOLIB.
13. When the installation and conversion of GN#LIB is complete, a list of ProGen Plus environment libraries will be displayed. Each active 6.0x environment should be selected for conversion. This process is described in more detail below.
14. Following the environment library conversion(s) the installation is complete.

## Post Installation Review

### **6.0x to 7.0x Upgrades: Converting Program Definition Environment Libraries**

If program definition environment libraries exist on your system, they will need to be converted to Release 7.0x format if you intend to use those program definitions with Release 7.0x. Environment libraries can be converted following the conversion of the product library GN#LIB. A panel will appear showing a list of environment libraries. Select the ones you wish to convert to release 7.0x. Prior to the conversion of each environment library, a save file called ENVLIBSAVF will be created in each environment library containing ProGen Plus release 6.0x environment files. When the conversion is complete a new set of files will exist in the environment library containing program definitions in Release 7.0x format.

A file containing the list of environment libraries and their conversion status exists in library GN#INST and can be maintained by program GN#ENV. The program GN#ENV is called by the conversion process to present these libraries for conversion. If you exit this program without completing the conversion of environment libraries, you can resume this process by calling GN#ENV (no parameters necessary). Ensure that the library GN#INST is in your library list when running GN#ENV.

After all environment libraries are converted, the entire installation and conversion process is complete.

### **To Sign on to Release 7.0x**

When the installation is complete you can sign on ProGen Plus as in the past, using the ZWRKPDF command. Any Designer IDs created with your previous release will still be available.

---

## Shell Source File Libraries

Each Designer ID references a shell source library. The default library will be GN#LIB. If you have changed the referenced library for this parameter it will be necessary to replace the shell source file GN#FSRC in this library with the current version of this file shipped with release 7.0x. Optionally you may wish to change this library reference back to GN#LIB. You can verify if any of your Designers are referencing different versions of the shells using option 2 from the list of designers presented by the ZWRKDSN command.

### Updating GN#FSRC in another library

If you have another library specified as the shell source library and you wish to replace that file with the 7.0x version then follow the steps below:

- Rename GN#FSRC in that library to another name.
- Execute the ZWRKDSN command, select 2=Change next to your designer ID.
- Press Enter through the panels. The ZWRKDSN command will check the shell source library you have specified for the file name GN#FSRC, when not detected a new copy of that file will be loaded from GN#LIB.

Note that if you have made custom modifications to the shells of your previous release, you will need to reapply these changes to your most current release. Use SEU to scan the shell source members for change dates to help with this process.

### Upgrading from ProGen ILE

If you are also upgrading ProGen ILE, follow the steps above but change the source file GN#FLES in addition to GN#FSRC.

---

## Reconciling Program Contexts

It is possible that you created another copy of the shell source file GN#FSRC because you wanted to add new program contexts to this file, or that you modified GN#FSRC and/or GN#GSRC in GN#LIB. If so, you should review the contexts in the latest release of ProGen Plus 7.0x prior to re-inserting those contexts into the shell source. Program contexts are added on an on-going basis.

The installation and conversion process is able to merge your 7.0x data file containing program contexts with the program contexts shipped with the latest release of 7.0x (records maintainable via the ZWRKPGMCTX command). It is not capable however, of merging context “tags” added to shell source members in the source files GN#GSRC or GN#FSRC. To do this you must review the 6.0x copy of members in these source files to determine which contexts you have added, and ensure that these contexts are added to the 7.0x shells if they are not already there.

### Upgrading from ProGen ILE

If you are upgrading ProGen ILE, follow the steps above but change the source file GN#FLES in addition to GN#FSRC and GNGLES in addition to GN#GSRC.

You can build (ZBLDPSD) and review (ZWRKPSD) program structure diagrams for each program style which will indicate the program contexts, and the subroutine they can be found in. See chapters 15 of the **ProGen Plus User's Guide** and the **Technical Reference Guide** for additional information.

---

## Reconciling Program Fields

It is possible that you created new program fields using the ZWRKPGMFLD command with your current release of 7.0x. The installation and conversion process is able to merge your existing data file containing program fields with the program fields file shipped with 7.0x. It is not capable however, of merging shell source additions and enhancements which may exist to support these new program fields. Many of the program fields are defined in a subroutine common to all program styles called \$PGMFS. This routine exists as a source member in the file **GN#LIB/GN#GSRC** with the member name **GN#SPGMF**. If new program fields were added, you may need to review this source member to determine if changes or additions were made. After the installation you can locate the 5.0x version of this member in **GN#LIBOLD/GN#GSRC**. Changes made to this member should be migrated to the same member in **GN#LIB/GN#GSRC**.

## Upgrading from ProGen ILE

If you are upgrading ProGen ILE, follow the steps above but change the source file **GN#GLES** in addition to **GN#GSRC**.

## Final Cleanup Tasks

Once you have successfully completed the steps outlined in this document, and you have a current backup, you can delete these objects, if still present on your system:

- GN#INST - Install library for previous releases (release 3.5 and prior)
- GN#LIB50 - Renamed Release 5.0x library
- GN#LIBOLD - Renamed Release 6.0x library or 5.x library (note library name given to rename your existing version of ProGen specified on the install panel)
- GN#I400 - Release 4.0x installation library
- GN#I450 - Release 5.0x installation library
- GN#OLD - pre-release 3.5 install library
- GN#DEMO - release 3.5 demo library
- GN#DEMO45 - release 4.5 demo library
- GN#ILEOLD – release 1.2x of ProGen ILE
- ENV40SAVF save files in each environment library - these contain release 4.0x definitions and can be deleted.
- ENV45SAVF save files in each environment library - these contain release 4.5x definitions and can be deleted.
- ENVLIBSAVF save files in each environment library - these contain release 6.0x definitions and can be deleted after confirming the installation of release 7.0x was successful.

## Additional Assistance

If you any questions concerning the installation of ProGen Plus Release 7.0x, please call or fax technical support at the following numbers:

Phone: 250-655-1766  
Fax: 250-655-1733  
E-mail: [excel@excelsystems.com](mailto:excel@excelsystems.com)

Additional, more detailed Technical Support information is available in the **Technical Support Guide**, which has been shipped to you with the product documentation

# Introduction: Overview of New Features

This New Features Guide for ProGen Plus 7.0 is divided into several sections, each describing the changes to different areas of the software.

## Knowledge Base

### Overview

This addresses one of our most commonly requested features over the years: the ability to be able to share Actions between definitions. The reason that this feature has been so long coming, is that if you copy Actions from one definition to another, a block of Actions could reference files, fields, work fields, condition, equations and values. And because part of our design philosophy in ProGen has been to protect you from getting compile errors, we were hesitant to add a feature to copy Actions, without being able to also copy all the other components.

The ProGen 7.0 KB addresses this issue by allowing you to export almost all components of your definitions to a shared area called the Knowledge Base, and from there, you can import to any definition. This is most important enhancement in Version 7, and we hope that you'll find it meets your expectations.

---

## Using the KB

The knowledge base is designed to allow users to store specific functions to later add to new definitions. These functions would become part of the KB and would be accessible from within any definition. The functions that are available are:

ACTIONS	Actions can be KB'ed individually or in groups.
WORKFLDS	All classes of work fields can be KB'ed.
EQUATION	Equations can be KB'ed.
COND	Conditions can be KB'ed (both free format and stranded)
VALUES	Values can be KB'ed.
PANEL	Panels and all attachments can be KB'ed.
DTAARA	Data areas can be KB'ed.

The items listed above when brought into the KB may contain other functions. For example, an action may contain references to a condition that contains references to a \*CONCAT work field that references a \*RESULT work field that has a data base field etc. This relationship must be preserved to make the KB item of any use when copied to a new definition. When any KB function is selected, the KB engine must determine all references that the function or any dependents of that function require.

Once this has been done a KB item can exist as an independent function. When a KB item is copied to an existing definition the KB engine must reverse engineer this item to exist within the predefined structure of the existing definition. To do this, all duplicate objects must be identified and resequenced where possible. This means all file relation and their dependents (@RS, @DL etc) must be resolved to the new layout, all work fields must be resequenced and resolved dynamically as the KB item is copied to the new definition, etc.

Once a function has been added to the KB it cannot be modified. You can view your existing entries using the command ZWRKKB. This command displays a list of all the KB entries that have been created, sorted by type. From the list you can use option 2 to change the description associated with the KB entry, and also add more detailed text for the entry using option 6. There are also additional options available for sorting and filtering the list.

---

## KB Technical Details

The KB entries are stored in the existing environment files. This means that if you were to copy a group of Actions, the KB entry for those actions would exist in the definition file GN#ACTF, within the environment library you're currently working in. The key for this new group of actions would be KB\_00001 in the program name key (ACPGM) the program library key (ACPLIB) would be blanks.

Groups are simply a listing of KB entries attached to a single group key. The group does not contain any KB entries it only references them.

Panels with actions attached will be placed after any existing panels. The associated actions will also be resequenced to match the new panel location.

If you pull a work field into a definition from the KB, and a work field with the same name already exists in the definition, the work field name will be resequenced to the first available @WKxxx name. This new name will be applied to all referencing items as they are pasted into the To definition. The renames will not be made to the originating KB entry.

Conditions, Values, Equations will all be resequenced to the next available number and that change will be applied to all referencing items as they are pasted into the To definition. The resequencing will not be made to the originating KB entry.

All files attached to the KB will be checked against the target definition and if the files exist in any form (External DS, Primary file, Aux file, Alt view) the file will not be added to the To definition. On adding a file that does not exist in the To definition, the file will be added to the next available sequence number within the type it was extracted from. The attached parm list (where applicable) will be retained but in many situations there will be no parm list. This is because the From definition may have been using this file as a Primary file and the To definition will almost always already have a Primary file. All @DS, @DL and @RS will be resequenced to match the new file position.

## Actions Editor/Work Field Features (Date/Time functionality ILE only)

Below are a list of features added to the Actions Editor and Work Fields, to better support Date/Time functions in ILE.

---

### \*DATTIM Work Fields

We have added a new work field class of \*DATTIM. This work field will allow users to create any valid date, time or timestamp data type. These work fields are similar to \*PANEL fields as there is no logic defined, they are simply declared.

\*DATTIM work fields also support the ability to define an initialization value, as do \*PANEL fields in 7.0. This value is validated for: correct separators, correct format and a correct date value.

This is a limited feature and is only supported in ILE RPG with a Conversion Option of \*NONE. If you are in RPG400 the \*DATTIM work field will not be displayed. If you are in ILE RPG and you create a \*DATTIM work field and message will be issued stating the Conversion Option has been changed to \*NONE. Throughout the PDA these fields will be displayed with their correct type: L, T and Z.

---

### New Date Actions

The DATINC, DATDIF and the DATDEC actions have been added to support handling Date, Time or Timestamp data types. These actions will support the functionality of the ADDDUR/SUBDUR operations in ILE.

#### DATINC

This action will allow you to add a duration to a Date/Time/Timestamp data field. The duration can be a constant, a numeric variable, in addition to a base date and can be added in any format \*YEARS, \*SECONDS etc...

**Date/Time** is optional and may contain a Date, Time or Timestamp field, subfield, array, array element, literal or constant. If factor 1 contains a field name, array or array element then its data type must be the same data type as the field specified in the result field. If the **Date/Time** is not specified the duration is added to the field specified in the result field.

**Duration Value** is required and a duration and may be a numeric field or constant with zero decimal positions. If the duration is negative then it is subtracted from the date.

**Duration Code** must be a valid duration code indicating the type of duration. The duration code must be consistent with the result field data type. You can add a year, month or day duration but not a minute duration to a date field).

**Result field** must be a date, time or timestamp data type field, array or array element. If Date/Time is blank, the duration is added to the value in the result field.

#### DATDIF

This action allows you to subtract a duration between two Date/Time/Timestamp fields and place the result in a numeric field in any format \*YEARS, \*SECONDS etc...

Date/time (1) is required and must contain a Date, Time or Timestamp field, subfield constant or literal.

**Date/time (2)** is required and must contain a Date, Time or Timestamp field, subfield constant or literal.

**Result duration** is the name of a zero decimal numeric field.

**Duration Code** contains a duration code denoting the type of duration. The result field will be negative if the date in factor 1 is earlier than the date in factor 2.

### **DATDEC**

This action will allow you to subtract a duration from a Date/Time/Timestamp data field. The duration can be a constant, a numeric variable, in addition to a base date and can be added in any format \*YEARS, \*SECONDS etc...

**Duration Value** is a numeric field or constant with zero decimal positions. If the field is negative then the duration is added to the field.

**Duration Code** must be a valid duration code indicating the type of duration. The duration code must be consistent with the result field data type. For example, you can subtract a year, month or day duration but not a minute

**From date/time** must be a date, time or timestamp data type field, array or array element. If factor 1 is blank, the duration is subtracted from the value in the result field.

**Result Date/Time** must be a date, time or timestamp data type field, array or array element. If factor 1 is blank, the duration is subtracted from the value in the result field.

### **Date Action Limitations**

The Date Actions are a limited feature in that they are only supported in ILERPG with a Conversion Option of \*NONE. If you are in RPG400 these actions will not be displayed. If you are in ILERPG and you create any of these actions a message will be issued stating the Conversion Option has been changed to \*NONE. All the actions have extensive validation that will stop compile errors from occurring.

The above features are based on an update we posted in 6.0 (PG60R051) that allowed the conversion type option to support \*NONE. When this option is \*NONE date/time/timestamp data types are handled as true date/time/timestamp data types. If the option is set to \*DATETIME (this is the default) then these data types are masked to alpha fields. ProGen 7 has built on this feature and will now display date/time/timestamp data types in their true format regardless of the conversion options. This means that a date data type will be displayed with a type of L not A as it is currently. This also extends to presenting the valid options available to this data type from various points in the product.

## Additional Design Options

The following features don't fall into any special category, but are generally resulting from specific client requests over the last while.

---

### Optionally Block Window Lookups on Output-only Fields

In previous releases of ProGen Plus, canny users had been known to change output-only fields using window lookups. While the cursor would never automatically go to a specific field, it was still possible to use the arrow keys or a mouse to place the cursor at the field location on the screen, and then press the Window function key to present the associate window lookup program, if any.

With release 7.0, based on the extended field attributes the window lookup functionality can be disabled. If the user defines N=Nondisplay, O=Output, X=Nondisplay/Output a lookup cannot be performed against that field while this condition exists (i.e. while the field is not input capable).

We have based this feature on a generator change that affects how windows are processed in general; they are no longer performed based on the cursor location, but on the returned field name from the DDS.

### Window Lookup Now Uses Field Name, not Location

Associated with the feature above, because the lookups no longer use the field location, you can also now have window lookups in multi-line, input capable subfiles (Styles E and N), where the lookup fields are directly above or below each other. Previously, because the field had occupied the same column position, only the first lookup would be executed.

---

### \*PANEL Work Fields Support Initialization Value

\*PANEL class work fields will now allow you to define an initialization value. This value will be defaulted into the \*PANEL on the initial running of the program. The initialization values is validated to only allow a default value less then or equal to the size of the \*PANEL field. Numeric fields will only allow entry of numeric data.

---

### Additional Validation on the Data and Time Data Type Functionality

If you change the Conversion Options from \*NONE to \*DATETIME a warning is issued stating that various aspects of your program will cause compile errors. The ability to select \*VARCHAR, \*GRAPHIC from the Conversion Options has been removed as they were not being supported.

---

### EVAL Action Supports Operation Extenders

The EVAL will now support operation extenders of H=Half adjust, M=Default precision rules, R="Result Decimal Position" precision rules. The valid options are shown below:

EVAL	Leave the field blank if no half adjust is to be performed, or if no extender is supported.
EVAL(H)	Type (H) if half adjust is to be performed.
EVAL(M)	Type (M) if the maximum number of digits rule is to be applied.
EVAL(R)	Type (R) if the result decimal position rule is to be applied: the precision of a decimal intermediate will be computed such that the number of decimal places will not be reduced by more than the number of decimal positions of the result of the assignment.

---

## Enhanced RPDA/PDA mobility

From the F4 functions list in the PDA and the RPDA there is a new option to go directly to work with conditions, equations and values.

---

## ZWRKDATCVT supporting ILE interface

ZWRKDATCVT added ability to toggle platform to view source from. A function key switches the source location allowing you to view the source member from GN#DLES (ILE) and GN#DSRC (RPG).

---

## AUTOINC Variable

When you define a primary file key a \*AUTOINC the value used to increment the numeric key is now a program field variable (\*INCVAL). By default this field is set to 1. To change the increment value you can update the \*INCVAL at \*INIT to an increment value of your choice.

---

## Optional Default PDA Text Placement

From the designer defaults (ZWRKDSN) you can select if field text for the flat panels is aligned to the right or left aligned immediately next to the field. If left alignment is chosen, the field is followed by a colon. If right aligned, a series of dots follow the field text.

**Right alignment:**

Customer number . . . . . 999999

**Left alignment to field:**

Customer number : 999999

---

## Default Designer ID Settings

You can now specify from the designer ID (ZWRKDSN) whether you want to enter the list of definitions (ZWRKPDF) with \*ALL (show the entire list) or the Designer ID limiting the list (as it works now). You can still toggle this display from the F11. To access this function run the ZWRKDSN then take an option 9=System defaults. On the 4<sup>th</sup> screen the 'Initial list of definitions' option can be set to \*ALL or left blanks.

---

## Direct Command Line Access

The PDA and the RPDA now have direct access to a command line using the F21=Command line function key. The Actions Editor also has a F21=Command line direct access.

---

## Partial Panel Redesigning

F15=Rebuild Panel now supports the ability to rebuild the specific panel the user is on or to rebuild all the panels. If only the current panel is rebuilt, the field set selected by the user will be placed on the panel. When rebuilding a single panel, if more fields are selected than can be placed automatically by the designer they are not included on the panel.

---

## Flat Panel Rebuilding Functionality Improvement

Styles D and H no longer attempt to auto-build the flat panel each time you press Enter from the subfile, if the user did not build the flat panel the first time through the design session. If the user decides to add a flat panel then the definition flow continues as normal.

---

## Reorganize F16 Alt view window

The alternate views window (F16 from the PDA) has been enhanced to display the files that the displayed keys are from. This allows users to distinguish which file the key they are placing is from.

---

## PDM Compliance in the ProGen Design Tool

Changes have been made to create a more uniform flow in the PDA. We have striven to make each panel in the design tool consistent across the entire product.

---

## PDM Compliance Issues for Generated Programs

The ability to optionally conform a definition to PDM cursor compliance has been added. This will force the cursor to remain on the subfile record that a subfile option was performed on when returning from the flat panel. This function has been added to the styles A, D, N and E. If you do not select this option then the cursor will return to the subfile control upon returning to the subfile from the flat panel. This option can be set at the designer level (ZWRKDSN).

With release 7, this option only affects the one function described above. In subsequent releases (and with interim updates) we plan to expand the list of functions affected.

---

## Direct Work Field and Actions Access

In the PDA function keys have been added to go directly to the actions editor (F18=Action ctx) and the work fields list (F17=Work fields). In the RPDA function keys have been added to go directly to the actions editor (F18=Action ctx) and the work fields list (F16=Work fields).

---

## \*RESULT Work Field Calculation Enhancements

We have addresses and issue with the calculation of result fields, and how the interim fields are handled. This resolves problems where the interim field truncates the value of the result. The interim fields are now created as 30,9. The interim field is then moved back to the result field by performing a MULT(H) \* 1 operation. This causes the interim field to be resolved correctly to the result field, and it's attributes.

---

## Read all Records Support

The FORRCASET action will now support \*LOVAL. This functionality will force all the records in a file to be read out side of a key list.

---

## Non-keyed File Support in Style V

The style V will now allow the entry of non keyed files. This will allow for processing entire files etc.

---

## Enhanced Designer Validations

Additional validation has been added to warn against removing all the EXIT function keys from the first panel/subfile of an interactive definition style.

---

## Improvements to the \*TOTAL processing

Corrected problem where the \*TOTAL based on a primary file field (A, E, N) was being totaled incorrectly when multiple records were deleted at once.

## Additional Report Control Fields

24 new report control fields have been added allow users more control over their reports. The report control fields will be available as system defaults and are now accessible from the ZWRKDSN subfile option 10. The old report control fields have been moved to this option as well and you will no longer have access to them from the 9=System defaults. All of the parameters are fully validated in the exact fashion that the OVRPRTF command.

The new report control fields are listed below:

### **Printer Device Type**                      **DEVTYPE**

The device type (DEVTYPE) parameter specifies the type of data stream created for a printer file. This parameter indicates whether the resulting data stream should be an intelligent printer data stream (\*IPDS), an SNA character stream (\*SCS), an ASCII data stream (\*USERASCII), an Advanced Function Printing data stream (\*AFPDS), line data (\*LINE), or mixed data (\*AFPDSLIN).

### **Measurement method**                      **PAGESIZE**

Specifies the length and width of the printer forms used by this printer file.

### **Front margin Offset down**                      **FRONTMGN**

#### **Front margin Offset across**

To use the margin parameters, the device type (DEVTYPE) parameter on the printer file must be \*AFPDS. For device types other than \*AFPDS, the AS/400 system calculates the margins.

Margins define the starting point of printed output on a piece of paper. The FRONTMGN parameter specifies the starting point on the front side of the paper; the BACKMGN parameter specifies the starting point on the backside of the paper.

There are two types of margins: front and back. Offset values, down and across, are used to fix the position of the margin. Across is defined as left to right. Down is defined as top to bottom.

Margins are measured in either inches or centimeters. The type of measurement is specified in the unit-of-measure (UOM) parameter on the printer file.

### **Back margin Offset down**                      **BACKMGN**

#### **Back margin Offset across**

To use the margin parameters, the device type (DEVTYPE) parameter on the printer file must be \*AFPDS. For device types other than \*AFPDS, the AS/400 system calculates the margins.

Margins define the starting point of printed output on a piece of paper. The FRONTMGN parameter specifies the starting point on the front side of the paper; the BACKMGN parameter specifies the starting point on the backside of the paper.

There are two types of margins: front and back. Offset values, down and across, are used to fix the position of the margin. Across is defined as left to right. Down is defined as top to bottom.

Margins are measured in either inches or centimeters. The type of measurement is specified in the unit-of-measure (UOM) parameter on the printer file.

### **Source drawer**                                      **DRAWER**

Specifies the source drawer used when automatic cut sheet feed mode is used (specified by FORMFEED (\*AUTOCUT)). When you specify a form definition, the value specified for the Drawer parameter will override the drawer value specified in the form definition. If you want to use the drawer value specified for the Form definition parameter, you must specify \*FORMDF for the Drawer parameter.

**Output bin****OUTBIN**

Specifies the destination bin of the output on printers that support multiple output bins. Possible values are:

**\*DEVD** The destination of the output is the default output bin for the printer device.

**Output bin** For printed output, specifies the output bin to be used on the printer device. Valid values range from 1 through 65535.

**Font identifier****FONT****Font Point size**

Specifies the font identifier and the point size (if the font supports multiple point sizes) of the font that is used for the following printers:

3130, 3812, 3816, 3820, 3825, 3827, 3829, 3831, 3835, 3900, 3912, 3916, 3930, 3935, 4028, 4224, 4230, 4234, 4247, 4312, 4317, 4324, 5219, InfoPrint 20, InfoPrint 32, InfoPrint 3000, and InfoPrint 4000 printers. This includes ASCII printers that emulate the 3812 or 5219 Printer.

**Degree of page rotation****PAGRRT**

Specifies the degree of text rotation for the following printers: 3160, 3130, 3812, 3816, 3820, 3825, 3827, 3828, 3829, 3831, 3835, 3900, 3916, 3930, 3935, 4028, 4312, 4317, 4324. It also specifies the degree of text rotation for the InfoPrint 20, InfoPrint 32, InfoPrint 3000, and the InfoPrint 4000. This parameter allows the user to specify the degree of text rotation on the page with respect to how you load the form into the printer.

**Print text****PRTTXT**

Specifies the printing of a line of text at the bottom of each page.

**Print on both sides****DUPLEX**

Specifies whether the output is printed on only one or on two sides of the paper. When you specify a Form definition, the value specified on the Duplex parameter will override the duplex value specified in the form definition. If you want to use the duplex value specified in the form definition, you must specify \*FORMDF for the duplex parameter.

**Front overlay****FRONTOVL****Front overlay library****Front Offset down****Front Offset across**

When an overlay is specified in a printer file, you can merge data from a spooled file onto the same piece of paper that the overlay is printed on. The FRONTOVL parameter specifies the overlay to be printed on the front side of the paper; the BACKOVL parameter specifies the overlay for the backside of the paper.

With your application program, you can use overlays that you created yourself using the Advanced Function Printing Utilities/400 licensed program, or overlays sent from System/390.

The merged document can be printed on any IPDS printer configured with AFP (\*YES) in the printer's device description.

The overlays can only be merged with a spooled file that was created using a device type (DEVTYPE) of \*SCS, \*IPDS, or \*AFPDS. The \*AFPDS must have been created on an AS/400 system.



## Additional RPDA Enhancements

In addition to supporting the report options shown above, a few other changes have been made to the Style F and the RPDA.

---

### Support Free format extended DDS keywords for external printer files.

From the RPDA you can attach extended DDS keywords at the field level. To access this function type \* beside the field displayed on the RPDA and press Enter; then, from the panel presented press F7=DDS keywords. This will take you to a list where single/multiple keywords can be attached. The keywords added are at the field level. This function will automatically select External Printer file usage. This function can be used to add underlines to fields or barcode keywords.

Any keyword available in a PRTF can be used. The entry of these keywords is completely free format and as a result any keying errors will result in the failed compile of the printer file. As well some keywords may conflict with the current setup of ProGen generated external printer files.

The entry of these keywords is similar to the keyword entry for fields in interactive program styles. There is no validation on what may be entered and the user may enter as many keywords as they need. When a keyword is defined a message is issued and the definition is changed to create an external printer file if not already selected.

---

### Supported DDS Keywords

Listed below is a sampling of DDS keywords associated with external printer files. The following list contains most of the DDS keywords that you can use to help you control how your printed output will look. Remember, some DDS keywords can only be used if the target printer supports them. For example: COLOR can only be specified if the target printer is a 4224.

For more detailed information on how to create and code DDS source files and to find out which printers support which DDS keywords, see the *DDS Reference* manual.

#### **ALIAS**

Alternative Name. Use this field-level keyword to specify an alternative name for a field. When the program is compiled, the alternative name is brought into the program instead of the DDS field name. The high-level language compiler in use determines if the alternative name is used. Refer to the appropriate high-level language reference manual for information about ALIAS support for that language.

#### **BARCODE**

Bar Code. Use this field-level keyword to print a field as a user-specified bar code. BARCODE is valid for IPDS printers. (A bar code is a pattern of bars of various widths containing data to be interpreted by a scanning device.)

#### **BLKFOLD**

Blank Fold. Use this field-level keyword for named fields that overflow onto subsequent print lines, to cause folding to occur at a blank rather than at the end of a line. If the blank fold keyword is not specified, the line folds at the end of the physical print line.

#### **CDEFNT**

Coded Font. Use this record- or field-level keyword to specify the coded font and point size for printing a named or constant field or fields within a record

#### **CHRID**

Character identifier. Use this field-level keyword to specify that a graphic character set and code page other than the device default can be used for this field. This can be important when extended alphabetic (characters such as u with an umlaut or c with a cedilla) are to be printed.

**CHRSIZ**

Character Size. Use this record- or field-level keyword to expand the width and height of a record or field. This applies to SCS double-byte characters, and IPDS and AFPDS single-byte characters.

**COLOR**

Color. Use this field-level keyword to specify the color for a field, if it is supported by the printer device. The COLOR keyword is used only by the 4224 printer. If you do not specify COLOR, or if the keyword is not valid for a printer device, black (the default value) is used.

**CPI**

Characters per Inch. This record- or field-level keyword specifies the horizontal printing density for the record format or field you are defining. Use CPI to:

- Darken logos and other printed graphics that you create using the DFNCHR keyword.
- Place more data in less space on printed forms.
- Fit the appearance of a form to your needs.

**CVTDTA**

Convert Data. This field-level keyword converts character data to hexadecimal data when the field is passed to the printer. You can use the CVTDTA keyword to define:

- Logos or emblems for a letterhead on your forms
- Alternative character sets or symbols (such as a copyright symbol)
- The appearance of a physical form (by adding vertical and horizontal lines that act as boundaries on the form or between positions on an invoice)
- IPDS bar code commands

**DATE**

Date. Use this field-level keyword to display the current job date.

**DATEFMT**

Date Format. Use this field-level keyword to specify the format of a date field.

**DATSEP**

Date Separator. Use this field-level keyword to specify the separator character for a date field.

**DFNCHR**

Define Character. The DFNCHR keyword allows you to define characters of your own design at the file or record level for the 5224 Printer and 5225 Printer. With this keyword you can specify DFNCHR more than once at the file or record level, or as many as 50 characters each time you specify DFNCHR.

**DFT**

Default. Use the DFT keyword to specify a constant value for constant (unnamed) fields.

**DLTEDT**

Delete Edit. Use this field-level keyword to specify that the OS/400 program is to ignore any edit code or edit word keywords specified for the referenced field. If a field description is referred to from a database file, DLTEDT prevents certain information from being referenced.

**DTASTMCMD**

Data Stream Command. Use record-level or field-level keyword to store a data stream command in a MODCA NOP structured field in the data stream of a spooled file. This information can be used to determine how to process a record or field on a particular page of a spooled file. DTASTMCMD keyword is only valid for printer files that have the device type specified as \*AFPDS.

**EDTCDE**

Edit Code. Use this keyword to edit output-capable numeric fields. An edit code is a letter or number indicating that editing should be done according to a defined pattern before a field is displayed or printed.

**EDTWRD**

Edit Word. If you cannot accomplish the desired editing by using the EDTCDE keyword, specify an edit word instead. An edit word is a user-defined word that specifies the form in which the field values are to print and clarifies the data by inserting characters, such as decimal points, commas, floating- and fixed-currency symbols, and credit balance indicators. Also use it to suppress leading zeros and to provide asterisk fill protection.

**FLTFIXDEC**

Floating-Point to Fixed Decimal. Use this field-level keyword to print a number in a floating-point field in fixed decimal notation.

**FLTPCN**

Floating-Point Precision. Use this keyword to specify the precision of a floating-point field.

**FNTCHRSET**

Font Character Set. Use this record-level or field-level keyword to specify the font and point size for printing a named or constant field or fields within a record.

**HIGHLIGHT**

Highlight. Use this record- or field-level keyword to indicate that a field should be printed in bold letters.

**IGCCDEFNT**

DBCS Coded Font. Use this record- or field-level keyword to print DBCS data contained in the AFPDS.

**INDARA**

Indicator Area. Use this keyword to remove option indicators from the buffer (also called the record area) and place them in a 99-byte separate indicator area.

**INDTXT**

Indicator Text. Use this file-, record-, or field-level keyword to associate descriptive text (indicating intent or use) with a specific indicator. You can specify INDTXT once for each indicator.

**MSGCON**

Message Constant. Use this field-level keyword to indicate that the text for a constant field is contained in a message description. If the message description does not exist at DDS compile time, the file is not created. If you change the message description, you must create the file again.

**PAGNBR**

Page Number. Use this field-level keyword to specify the location of an unnamed, four-digit, zoned decimal field to contain the page number.

**POSITION**

Position. Use this field-level keyword to specify the position of a field using the units specified on the unit of measure (UOM) printer file parameter.

**PRTQLTY**

Print Quality. Use this record- or field-level keyword to vary the print quality within the file. The PRTQLTY keyword is allowed only on records or fields for which a CHRSIZ or BARCODE keyword applies.

**REFFLD**

Referenced Field. Use this field-level keyword when referring to a field under one of these three conditions:

- The name of the referenced field is different from the name in positions 19 through 28.
- The name of the referenced field is the same as the name in positions 19 through 28, but the record format, file, or library of the referenced field is different from that specified with the REF keyword.
- The referenced field occurs in the same DDS source file as the referencing field.

**SKIPA**

Skip After. Use this file-, record-, or field-level keyword to specify that the printer device is to skip to a specific line number after it prints one or more lines.

**SKIPB**

Skip Before. Use this file-, record-, or field-level keyword to specify that the printer device is to skip to a specific line number before it prints the next line(s).

**SPACEA**

Use this record- or field-level keyword to specify that the printer device is to space some number of lines after it prints one or more lines.

**SPACEB**

Space Before. Use this record- or field-level keyword to specify that the printer device is to space some number of lines before it prints the next line or lines.

**TEXT**

Text. Use this record- or field-level keyword to supply a text description (or comment) for the record format or field.

**TIME**

Time. This field-level keyword prints the current system time.

**TIMFMT**

Time Format. Use this field-level keyword to specify the format of a time field.

**TIMSEP**

Time Separator. Use this field-level keyword to specify the separator character used for a time field.

**TRNSPY**

Transparency. This field-level keyword prevents code points you have redefined (using the DFNCHR keyword) from being interpreted as SCS printer control commands when your program sends an output operation that prints the field you are defining.

**TXTRTT**

Text Rotation. Use this field-level keyword to specify the rotation of the text on a page.

**UNDERLINE**

Underline. Use this field-level keyword to specify that the OS/400 program is to underline the field when it is printed.

---

**\*VALUE Work Fields Displaying Values While in the RPDA**

If you place a \*VALUE work field on the RPDA the first value unconditionally will be displayed on the RPDA. This replaces a series of “OOOO...” or “6666...” with the first value attached to the \*VALUE work field. The value will be display regardless of conditioning that may be attached. This feature allows users to have a visual representation of how their report may look.

This is especially useful if users are placing text constants using \*VALUE fields to add text to your report, to which you want to apply DDS keywords (DDS keywords can't be applied to plain text).

---

**Blank After Option on Reports**

To access the blank after feature do a \* prior to the field you wish to assign the blank after to. The blank after is used to reset a numeric field to zeros or a character field to blanks. Resetting fields to zeros is useful when totals are accumulated and written for each control group in a program. After the total is accumulated and written for one control group, the total field can be reset to zeros before accumulation begins on the total for the next control group.

If blank after is specified for a field to be written more than once, the blank after should be entered on the last line specifying output for that field, or else the field named may be set to blanks or zeros after the field is written to the output record.