

## **APL2: What's New**

**Michael T. Wheatley**

**IBM .  
APL Development  
General Products Division  
Santa Teresa Laboratory  
San Jose, California**

---

## **Abstract**

APL2 Release 3 provides substantial new and changed function in comparison to previous releases of APL2. This document itemizes the new features and changes, provides a brief description of each and directs the user to the appropriate APL2 Release 3 publications which contain additional information. The new and changed function is presented in no specific order.

---

# Contents

Support of the IBM 3090 Vector Facility	1
Packaged Workspaces	2
DBCS (Kanji) Support	3
Performance Monitoring Facility	4
APL2 for Prewritten Applications	5
New Primitive Functions	6
Selective Specification Enhancements	7
Interpreter Performance Enhancements	8
System Editor Enhancements	9
Processor 11 Enhancements	10
External Functions Enhancements	11
Shared Variable Processor Enhancements	12
AP 121 Enhancements	13
AP 127 Enhancements	14
Workspace Enhancements	15
National Language Files	16
APL2 Publications Enhancements	17
Minor Enhancements - Interpreter	18
Minor Enhancements - Executor	19
Minor Enhancements - Auxiliary Processors	20
Prerequisite Products	21



---

## Support of the IBM 3090 Vector Facility

**Description:** When invoked on an IBM 3090 CPU with the Vector Facility, APL2 will automatically recognize the presence of that facility and use it to improve the performance of APL applications.

A number of the APL primitive scalar functions and operators have been recoded to utilize the 3090 Vector Facility if it is available. Further, the APL syntax analyzer has been extended to recognize strings of scalar operations in an APL expression which can be executed using the 3090 Vector Facility and to build optimized loops to perform the necessary execution.

In APL2 Release 3, the following primitive functions, operators and idioms will utilize the 3090 Vector Facility if it is available and if the argument data is the correct type and of sufficient size to use the facility to advantage:

- conjugate, negative, direction, natural logarithm, reciprocal, exponential, Pi times, magnitude, square root
- addition, subtraction, multiplication, division, minimum, maximum, power, relational functions (<, ≤, =, ≥, >, ≠) circle functions (<sup>-</sup>12, <sup>-</sup>11, <sup>-</sup>10, <sup>-</sup>6, <sup>-</sup>4, 0, 1, 2, 3, 4, 9, 10, 11, 12)
- +/ boolean vectors, +.x, +/, r/, l/, expand, general reduction, general inner product, v1r/v, v1l/v, +/l, r/l, l/l.

Not all cases of these primitives utilize the Vector Facility: in general, cases with real arguments are handled, some cases with integer and complex arguments are also handled.

No modification to existing APL applications is required to use these facilities. Use of the 3090 Vector Facility can be disabled by specifying the invocation option SYSDEBUG(16).

**Reference:** Since these facilities do not affect the externals of the APL language or system, no detailed documentation is provided in other APL2 publications.

For a general discussion of the IBM 3090 processor, the IBM 3090 Vector Facility, and performance considerations, see the appropriate articles in the IBM Systems Journal, Vol. 25, No. 1, 1986 (G321-0082).

---

## Packaged Workspaces

**Description:** With APL2 Release 3, workspaces may be encapsulated or "packaged" and converted to load modules. Objects in such packaged workspaces can be accessed dynamically using `□NA`.

Packaged workspaces are loaded dynamically when objects in them are accessed and may be placed in the LPA in MVS or in a DCSS in VM and thus may be shared on a read only basis between multiple simultaneous APL users.

Packaged workspaces each contain their own "name scope". That is to say, names in packaged workspaces do not conflict with names in the user's workspace or in other packaged workspaces. Thus, packaged workspaces provide an attractive means by which APL applications may be implemented, combined with other applications, and provided to users.

A number of packaged workspaces are provided with APL2:

- *TIME*: A set of functions which allows timing of applications and provides performance information to application developers.
- *PACKAGE*: A set of functions which allows saved workspaces to be converted to packaged workspaces.
- *DFMT*: A set of functions which performs aligned formatting of arrays containing DBCS (Kanji) data.
- *CMSIVP* and *TSOIVP*: Installation verification functions for CMS and TSO.

**Reference:** APL2 System Service Reference manual, Chapter 24.

---

## DBCS (Kanji) Support

- Description:** The APL system and interpreter have been extended to provide support of DBCS data on devices such as the IBM 5550 which allow display of DBCS characters. These extensions are substantial and affect a number of the components of the APL2 system:
- The APL2 Session Manager has been extended to support input, display, logging and copying of DBCS character data. DBCS data may also be included in certain Session Manager commands (eg: PFK); AP 120 has also been extended to provide similar facilities.
  - The APL2 Interpreter has been extended to allow input and output of DBCS data which is mapped to and from extended character data in the APL workspace. Extended characters consume 4 bytes per character in the APL workspace, are distinguishable from normal APL (1-byte) characters, and may be processed by APL functions and operators which process character data.
  - The APL interpreter has been extended to allow character literals (character strings within quotes) and comments to contain extended characters.
  - The APL interpreter has been extended to provide consistent handling of extended character data in primitives (with the exception of dyadic grade and picture format) and system functions which process character data.
  - The commands `)IN`, `)OUT`, `)COPY` and `)FCOPY` have been extended to handle objects containing extended character data.
  - The APL editors (Editor 1, Editor 2 and the interface to system editors) have been extended to allow editing of objects which contain DBCS data.
  - AP 127 (DB2, SQL/DS) has been extended to support DBCS data in SQL statements and with the GRAPHIC data types.
  - The AP 110, 111 and 210 (DBCS option has been improved to provide consistent handling of DBCS data.
  - The external functions `CTK` and `KTC` have been extended to provide consistent handling of DBCS and mixed SBCS/DBCS data. These functions provide a means to convert between extended character data and mixed SBCS/DBCS data.
  - A new external function, `DFMT`, is provided to facilitate formatting and display of arrays containing DBCS data.
  - The national language support for APL messages has been extended to allow DBCS characters in messages. A new set of translations for interpreter messages into Japanese is provided and may be accessed by specifying `ENVIT←'JAPANESE'`.
  - A new invocation option, DBCS, is provided to allow or disable access to many of these facilities.
- Reference:** Because these new facilities are extensive and affect many of the components of the APL2 system, there is no single source of reference for them. The Language Reference, the System Services Reference, and Using the Supplied Routines manuals provide information on most of the enhancements in their appropriate sections.

---

## Performance Monitoring Facility

**Description:** The external function *TIME* provides a mechanism by which performance information can be gathered and reported for APL applications. CPU time and number of times executed can be monitored either by line or by program. Using this facility, application developers can isolate "hot spots" in APL applications and use this information to make decisions concerning systems design and optimization of applications.

**Reference:** Using the Supplied Routines manual



---

## APL2 for Prewritten Applications

**Description:** In addition to the full APL2 system, a subset of the APL2 system is available as a separate product, APL2 Application Environment (5668-003). This subset is designed to allow pre-written APL applications to execute, but precludes the use of APL2 program development facilities. Installations may wish to provide this subset of APL2 to application users and restrict access to the full APL2 system to application developers.

This subset of APL2 provides all of the APL2 language and system facilities with the exception of the following:

- No terminal support or Session Manager support is provided. All terminal I/O must be performed using auxiliary processors such as AP 126 (GDDM) or AP 317 (ISPF), or using the APLIN/APLPRINT ddnames.
- No saved workspaces or workspace libraries are provided or available. All applications are expected to be provided as packaged workspaces. Since saved workspaces are not supported, the commands `)LOAD`, `)SAVE`, `)CLEAR`, `)LIB`, `)WSID`, `)DROP`, `)COPY`, `)PCOPY`, and `)MCOPY` are not supported.
- The commands `)IN` and `)OUT` are not supported.
- No editors are provided; the `)EDITOR` command is not supported.

**Reference:** The installation manuals describe installation of the full APL2 system as well as this subset of the APL2 system.

---

## New Primitive Functions

**Description:** Four new primitive functions are provided in the APL language:

- **Index** and **Index with Axis** provide a functional form of cross-sectional indexing of APL arrays and provide an alternative to the traditional Bracket Indexing syntax. These new indexing functions can be used on the left of specification and in conjunction with the Each operator to provide more powerful indexing constructs.
- **Partition** and **Partition with Axis** provide the ability to "partition" an array at calculated positions into an array of vectors.

**Reference:** Language Reference manual

---

## Selective Specification Enhancements

**Description:** Selective specification provides a powerful facility which allows portions of an array to be replaced. In previous releases, the system has been limited in the types of selection operations which could be specified on the left of specification. With APL2 Release 3, many of these restrictions have been lifted and a larger set of operations, including 'enlist' and 'pick each enclose', can be used in selective specification.

**Reference:** Language Reference manual

---

## Interpreter Performance Enhancements

**Description:** The performance of a number of primitive functions and operators has been improved:

- pick
- first N drop
- catenate with rank > 1
- vector notation
- compress/replicate
- expand
- match
- membership, index of, without for fullword integer arguments
- membership, index of, without for vector of character vectors arguments
- certain cases of logarithm, exponential, magnitude, power and circle functions
- most primitive scalar functions with scalar arguments and several commonly used scalar functions with depth 1 arguments

**Reference:** Since these enhancements involve performance improvements in the internal operation of APL2 and no changes to the language definition, no documentation is provided.

---

## **System Editor Enhancements**

**Description:** The interface to system editors (e.g., XEDIT or ISPI) has been enhanced to allow any array of rank 2 or less to be edited. This includes nested arrays and arrays containing numeric data.

**Reference:** Language Reference manual, System Services manual.

---

## Processor 11 Enhancements

**Description:** Processor 11 has been enhanced to:

- Allow access to objects in packaged workspaces
- Allow the use of multiple private NAMES files
- Allow Fortran functions (in addition to Fortran subroutines) to be executed. This support may also be used in Assembler Language programs to return explicit results.
- Allow arrays up to depth 181 to be passed as arguments to external functions.
- Provide additional facilities for external functions using `:LINK.FUNCTION` including the ability for those functions to execute APL expressions and reference or specify named APL variables. These external functions may now also have an associated execution environment routine (`:INIT. tag`).
- Support for ESSL (the Engineering and Scientific Subroutine Library, program number 5668-863). A NAMES file for ESSL routines will be provided with ESSL and documentation on calling these routines is provided in an APL publication.

**Reference:** System Services Reference manual; Using the Supplied Routines manual.

---

## External Functions Enhancements

**Description:** A number of the existing external functions, distributed with APL2, have been enhanced and a number of new external functions are provided:

- *CTK* and *KTC* have been extended to provide improved and consistent DBCS support.
- *ATR*, *RTA* and *PFA* have been extended to handle patterns and arrays up to depth 181.
- A new external function, *OPTION*, is provided to allow query or setting of the invocation options *CASE*, *DBCS*, *DEBUG*, *QUIET*, *SYSDEBUG*, and *TRACE*.
- A new external function, *MSG*, is provided to allow the user to generate, display, queue or check APL messages.
- A new external function, *EXP*, is provided to allow reference or specification of a named variable in the previous name scope, or to execute a named function in the previous name scope. This function is designed to be used in applications running in packaged workspaces.
- A new external function, *QNS*, is provided to query the current name scope. This function is designed to be used in applications running in packaged workspaces.
- Documentation comments have been removed from Processor 11 NAMES files distributed with APL2 and placed in the Using the Supplied Routines manual. This change will improve the performance of *QNA* for external routines distributed with APL2.

**Reference:** The external functions provided with APL2 are documented in the Using the Supplied Routines Manual.

---

## Shared Variable Processor Enhancements

**Description:** A new interface protocol is provided for auxiliary processors to provide improved facilities for AP's designed to interface with APL2 and to allow the implementation of local and global AP's. Local AP's are single servers designed to interface with a single APL user and reside in the APL user's address space or virtual machine. Global AP's may be multi-servers designed to interface simultaneously to multiple API. users; they typically reside in a separate address space or virtual machine.

In the MVS environment, the global SVP has been enhanced to recognize unexpected termination of an address space and to sign the user off the global SVP if necessary. As a result of this enhancement, the global SVP signon installation exit (ISFCNAME = ) is no longer required or supported. The global SVP initialization installation exit (GSECNAME = ) continues to be supported.

The APL2 Release 3 global SVP must be used with API.2 Release 3; it supports sharing of variables between Release 3 users and users of earlier releases of APL2.

**Reference:** The new AP interface protocol is documented in the Processor Interface Reference manual. The global SVP installation exit is documented in the Installation and Customization under TSO manual.



---

## AP 121 Enhancements

**Description:** Auxiliary processor 121 (APL file I/O) has been enhanced to allow arrays of arbitrary size (4  $\square AT$ ) to be read or written using direct files. Previously, arrays were restricted to 4054 bytes in size when read or written on AP 121 files. With this enhancement, array size is limited only by workspace size, shared memory size and freespace size.

A new service request, *Q* (Query), has been added to AP 121 to allow users to obtain the names of files stored in AP 121 libraries.

**Reference:** System Services Reference manual.

---

## AP 127 Enhancements

**Description:** The following enhancements are provided for AP 127 (DB2, SQL/DS):

- Support for DBCS data in SQL statements and using GRAPHIC, VARGRAPHIC, and LONG VARGRAPHIC data types.
- Support for dynamic isolation level switching (via a new AP 127 command, ISOL).
- Support for DATE, TIME and TIMESTAMP data types.
- Support for the REAL (single precision floating point) data type (DB2 only).
- Extension of the AP 127 DESCRIBE command to provide labels and null information.
- Return of null prototypes when the result table is empty on FETCH.
- Availability of query cost information after a PREP command.

**Reference:** System Services Reference manual, Using SQL manual-

---

## Workspace Enhancements

**Description:** A new workspace, *CHARTX*, is provided to facilitate access to the GDDM Interactive Chart Utility.

The *CMSIVP* and *TSOIVP* workspaces are no longer provided. They have been replaced with packaged workspaces containing functions of the same name.

**Reference:** Using the Supplied Routines manual

---

## National Language Files

- Description:** A set of national language definition files is now distributed with APL2. These files replace the previously hard-coded national language translations. Installations may extend these files to provide command and message translations for additional languages, to provide translations for additional messages, or to improve translations or adjust for local dialects.
- Reference:** Installation and Customization manuals. Additional documentation is provided in the sample APL2LANG files distributed with APL2.

---

## APL2 Publications Enhancements

**Description:** All APL2 publications, with the exception of the following, have been updated and improved:

- An Introduction to APL2
- APL2 Graphpak: User's Guide and Reference
- APL2 Programming Guide

Notable improvements include:

- A new publication, APL2 Processor Interface Reference, is provided to document the new auxiliary processor interface protocol. In addition to this information, this manual also includes information on writing external functions with `:LINK.FUNCTION` which was previously contained in the System Services Reference manual.
- The Using the Supplied Workspaces manual has been renamed to Using the Supplied Routines and extended to include information on external functions and packaged workspaces distributed with APL2. Finally, this manual also includes information on calling ESSL routines through Processor 11.
- The Diagnosis Guide and Diagnosis Reference manuals have been combined to form a new manual: APL2 Diagnosis.
- The Language Reference manual has been reorganized and resequenced to provide better access to information.
- The user response suggestions in the Messages and Codes manual have been improved.

**Reference:** APL2 publications.

---

## Minor Enhancements - Interpreter

**Description:** The following minor enhancements have been made to the APL2 interpreter:

- The `)SI`, `)SINL` and `)SIS` commands have been extended to allow an optional integer argument. When specified, this argument limits output to the specified number of stack levels.
- A new command, `)SIC`, has been added to conform to the new APL ISO standard. It provides the same function as the `)RESET` command.
- A number of `)MORE` messages have been added for language and system command errors.
- The left argument of dyadic `⊞NA` has been changed to allow specification of the associated processor number and an arbitrary array which is passed to the associated processor. Processor 11 interprets the array passed as a name class, the name of the names file, or the load library and member name for a packaged workspace.  
Monadic `⊞NA` and `2 ⊞TF` have been changed to reflect the change in dyadic `⊞NA`.
- Monadic and dyadic `⊞NA` have been extended to allow surrogate names which begin with `⊞`. This change allows users to access system functions and system variables in packaged workspaces. `2 ⊞TF` has also been modified to reflect this change.

**Reference:** Language Reference manual

---

## Minor Enhancements - Executor

**Description:** The following minor enhancements have been made to the APL2 executor:

- A new invocation option, `RUN`, is provided to allow users to specify the name of an external function (typically in a packaged workspace) to be run when APL2 is invoked.
- Installations may now provide a set of "greeting" messages to be displayed when APL2 is invoked.
- In the MVS environment, the maximum library number for SAM public libraries can now be specified by an installation. It was previously fixed at 999.
- In the MVS environment, installation exit routines have much more control over dataset naming conventions for SAM workspace libraries.
- In the MVS environment, the interface to the installation exit has been extended and fully documented.
- In the MVS/XA environment, the Session Manager has been moved above the 16MB line to provide additional virtual storage constraint relief.
- In the MVS environment, `SYSDEBUG(64)` can now be set and reset dynamically with the `)CHECK SYSTEM` command or the external function `OPTION`.
- The command `)CHECK SYSTEM SM(ON|OFF|TRY)`, previously available only in VM/CMS, is now also provided in the MVS environment.
- In the MVS environment, APL2 will now dynamically determine terminal type (`TERMCODE`), if it is not specified, for 3270 terminals other than the 3277.
- In the VM/CMS environment the default `FREESPACE` size has been changed from 0 to 256K.

**Reference:** System Services Reference manual, Installation and Customization manuals, Diagnosis Reference manual

---

## Minor Enhancements - Auxiliary Processors

**Description:** The following minor enhancements have been made to auxiliary processors:

- AP 100 (MVS only) now permits authorized commands and CLISTS containing authorized commands to be invoked from APL2. The same support is also provided with the command *)HOST*.
- AP 100 (MVS only) the built-in command *)USER* has been improved to obtain system, subsystem, and APL2 product names and levels dynamically.
- Auxiliary Processor 111 (CMS and TSO) has been extended to allow matrices to be written with a single specification of the data variable. Rows from matrices so written will be written to the file as multiple sequential logical records.
- The AP 120 *LINE N* command has been extended to return the contents of the specified line from the Session Manager log in the data variable.

**Reference:** System Service Reference manual



---

## Prerequisite Products

**Description:** The following changes have been made to products that are prerequisite to APL2 Release 3:

- VM/SP Release 4 or VM/SP IIPO Release 4.2 are now prerequisite to the use of APL2 in the VM/SP environment.
- APL2 Release 3 will also operate under VM/XA SP Release 1 in 370 compatibility mode.
- TSO/E is now prerequisite to the user of APL2 in the MVS environment.
- GDDM Version 2 is required for DBCS support on the IBM 5550.
- SQL/DS Version 1, Release 3 or later and DB2 Version 1, Release 2 or later is now required for use of AP 127.

**Reference:** APL2 General Information manual.





