

Release Notes
RTS Version 3.9.14 (19991003)
03 October 1999

1.0 Overview

This document describes changes included within:

Module Name	Module Description
-----	-----
SysConf	System Configuration
TPS	TYX Programming Support
PORF	PAWS Output Report Formatter
Targetter	ATLAS Program Targetter
WinUtil	TYX MS Windows Utilities
PLI	PAWS LAPS Interpreter
PTE	PAWS Test Executive

Note SysConf is used by TPS, PORF, Targetter, WinUtil, PLI and PTE.
 TPS is used by PORF, Targetter, WinUtil, PLI and PTE.
 WinUtil is used by PLI and PTE.

Module	Changes	Rebuilt	Current Version
-----	-----	-----	-----
SysConf	None	N/A	19990610
TPS	None	No	19990803
PORF	None	No	19990610
Targetter	None	No	19990610 3.8.3
WinUtil	None	No	19990610
PLI	None	No	19990705 3.9.9
PTE	Major	Yes	19991003 3.9.14

PTE means both the Run-Time System (RTS) and the Simulator (SIM).

PTE, RTS, CEM and SIM mean all Platforms.
 UPTE, URTS, UCEM and USIM mean UNIX Platforms only.
 WPTE, WRTS, WCEM and WSIM mean MS-Windows Platforms only.
 XPTE, XRTS and XSIM mean X-Window Platforms only.

NOTICE TO CEM USERS

This version of the RTS is compatible with CEM Modules built with CEM Files distributed with RTS Version 19980714 3.9.7 and later. You may install this version of the RTS without rebuilding your CEM Module. However, in order to make use of the new CEM capabilities, you **MUST** install this version of the RTS, and then you **MUST** recompile and relink your CEM Module(s) with the TYX-supplied CEM Files distributed with this version of the RTS.

As time goes on, RTS/CEM testing to provide error-free backwards-compatibility becomes increasingly difficult. Therefore, TYX Corporation very strongly recommends that CEM Users recompile and relink their CEM Modules after installing a new version

of the RTS.

- 1.1 Enhancements
 - 1.1.1 System Configuration – None.
 - 1.1.2 TYX Programming Support - None.
 - 1.1.3 PAWS Output Report Formatter - None.
 - 1.1.4 ATLAS Program Targetter - None.
 - 1.1.5 TYX Windows Utilities - None.
 - 1.1.6 PAWS LAPS Interpreter - None.
 - 1.1.7 PAWS Test Executive - Major.
 - RTS-to-CEM Statement Information Message:
 - Current Noun and Measured Characteristic.
 - CEM Group Modifiers.
- 1.2 Problem Reports - None.

- 2.0 Detailed Description
 - 2.1 Enhancements
 - 2.1.1 System Configuration - None
 - 2.1.2 TYX Programming Support - None.
 - 2.1.3 PAWS Output Report Formatter - None.
 - 2.1.4 ATLAS Program Targetter - None.
 - 2.1.5 TYX Windows Utilities - None.
 - 2.1.6 PAWS LAPS Interpreter - None.

2.1.7 PAWS Test Executive - Major.

2.1.7.1 RTS-to-CEM Statement Information Message.

Prior to this release, the RTS-to-CEM Statement Information Message did not contain the Current Noun and Measured Characteristic. The CEM Kernel acquired the Current Noun and Measured Characteristic from the RTS-to-CEM FNC Message and retained it for subsequent use by a CEM Device Driver. However, RTS-to-CEM FNC Messages are not generated for several Single Action ATLAS Statements (e.g., CONNECT, REMOVE, etc.), which resulted, under certain conditions, in an incorrect Current Noun and/or Measured Characteristic being returned to the CEM Device Driver.

With this release, the RTS-to-CEM Statement Information Message contains the Current Noun and Measured Characteristic.

2.1.7.2 CEM Group Modifiers.

Note Although CEM Group Modifier capability is and will be available to all CEM Device Drivers, Group Modifiers themselves are "defined" by a particular ATLAS Compiler and are peculiar to a specific ATLAS Subset / Station. Therefore, this section provides only a very general description of the CEM Group Modifier capability. For those Customers whose ATLAS Compiler "defines" Group Modifiers, a full description will be provided in a separate document. For all other Customers, this section contains no relevant information for CEM Device Drivers.

Prior to this release, the CEM Kernel did not support Group Modifiers.

With this release, the CEM Kernel supports Group Modifiers. This capability has been provided to allow ATLAS Compilers and CEM Device Drivers to more easily deal with ATLAS Statements that contain multiple references to the same Modifier within a single ATLAS Statement.

A Group Modifier is a Modifier that has one or more lists of other Modifiers attached to it. These lists are called Group Modifier Lists. In addition, Group Modifiers may be nested (i.e., a Group Modifier List may itself contain one or more Group Modifiers). As currently implemented:

- There is no limit on the number of Group Modifier Lists that a Group Modifier may have.
- The maximum level of Group Modifier nesting is three.

To support Group Modifiers:

- Modifier Data Structure MODDAT has been expanded to hold Group Modifier information.
- New CEM Macro Group (Group Modifiers) containing eight Group Modifier CEM Macros has been added to allow CEM Device Drivers to access Group Modifier data.

Refer to Section 3.0 for a full description of these new CEM Macros.

- CEM User Stub Functions have been enhanced to print Group Modifier information.

2.2 Problem Reports - None.

3.0 CEM Help

New CEM Macro Group - Group Modifiers

```
SetGroupModsON()  
SetGroupModsMOD()  
SetGroupModsLEVEL()  
SetGroupModsGROUP()  
SetGroupModsOFF()  
  
GetCurGroupModMOD()  
GetCurGroupModLEVEL()  
GetCurGroupModGROUP
```

Name: SetGroupModsON

Type: Group Modifiers

Usage: `int nStatus;`
`nStatus = SetGroupModsON(M_XXXX);`

Description: This Macro enables Group Modifiers processing at Group Level 1 for Group #1 of Group Modifier **M_XXXX**. If the operation is successfully completed (and until otherwise directed), all *GetDatum()* requests will search for Datums in the first Group Modifier List of Group Modifier **M_XXXX** at Group Level 1 (instead of searching the Modifier List associated with the Current Device / Channel). This Macro must be called before any of the other Group Modifiers Macros are called.

Result Current Group Modifier = **M_XXXX**
Current Group Level = 1
Current Group Number = 1

If an error was detected, a negative number (currently -1) is returned.

Errors are:

- **M_XXXX** not valid.
- No Current Device / Channel.
- Group Modifiers capability not available.
- No Group Modifiers defined.

If the operation could not be performed, a zero (0) is returned. Failures are:

- Group #1 at Group Level 1 for Modifier **M_XXXX** not found.

If the operation was successfully completed, a positive number (currently +1) is returned.

See Also: *SetGroupModsMOD()*, *SetGroupModsLEVEL()*, *SetGroupModsGROUP()*, *SetGroupModsOFF()*, *GetCurGroupModMOD()*, *GetCurGroupModLEVEL()* and *GetCurGroupModGROUP()*

Name: SetGroupModsMOD

Type: Group Modifiers

Usage: `int nStatus;`
`nStatus = SetGroupModsMOD(M_XXXX);`

Description: This Macro advances Group Modifiers processing to the next Group Level for Group #1 of Group Modifier **M_XXXX**. If the operation is successfully completed (and until otherwise directed), all *GetDatum()* requests will search for Datums in the first Group Modifier List of Group Modifier **M_XXXX** at the new Group Level.

Result Current Group Modifier	= M_XXXX
Current Group Level	= Previous Group Level + 1
Current Group Number	= 1

This Macro may be thought of as a Subroutine Call. The Macro *SetGroupModsLEVEL()* may be thought of as the matching Subroutine Return.

If an error was detected, a negative number (currently -1) is returned. Errors are:

- **M_XXXX** not valid.
- No Current Device / Channel.
- Group Modifiers capability not available.

If the operation could not be performed, a zero (0) is returned. Failures are:

- Group Modifiers processing not enabled.
- Group #1 at next Group Level for Modifier **M_XXXX** not found.

If the operation was successfully completed, a positive number (currently +1) is returned.

See Also: *SetGroupModsLEVEL()* and *GetCurGroupModLEVEL()*

Name: SetGroupModsLEVEL

Type: Group Modifiers

Usage: `int nStatus;`
`nStatus = SetGroupModsLEVEL();`

Description: This Macro backs-up Group Modifiers processing to the Group Level and Group Number of the Group Modifier that was current when the most recent call to *SetGroupModsMOD()* was called.

Result

Current Group Modifier	= Restored
Current Group Level	= Restored
Current Group Number	= Restored

This Macro may be thought of as a Subroutine Return for a corresponding *SetGroupModsMOD()*.

If an error was detected, a negative number (currently -1) is returned.

Errors are:

- No Current Device / Channel.
- Group Modifiers capability not available.
- No Group Modifiers defined.

If the operation was successfully completed, a positive number (currently +1) is returned.

See Also: *SetGroupModsMOD()* and *GetCurGroupModLEVEL()*

Name: SetGroupModsGROUP

Type: Group Modifiers

Usage: `int nStatus;`
`nStatus = SetGroupModsGROUP();`

Description: This Macro advances Group Modifiers processing to the next Group Modifier List at the Current Group Level for the Current Group Modifier . If the operation is successfully completed (and until otherwise directed), all *GetDatum()* requests will search for Datums in the next Group Modifier List of the Current Group Modifier at its Current Group Level.

Result

Current Group Modifier	= No change
Current Group Level	= No change
Current Group Number	= Previous Group Number + 1.

If an error was detected, a negative number (currently -1) is returned. Errors are:

- No Current Device / Channel.
- Group Modifiers capability not available.

If the operation could not be performed, a zero (0) is returned. Failures are:

- Group Modifiers processing not enabled.
- No more Groups for Current Group Modifier at its Current Group Level.

If the operation was successfully completed, a positive number (currently +1) is returned.

See Also: *SetGroupModsON()*, *SetGroupModsMOD()*, *SetGroupModsLEVEL()* and *GetCurGroupModGROUP()*

Name: SetGroupModsOFF

Type: Group Modifiers

Usage: `int nStatus;`
`nStatus = SetGroupModsOFF();`

Description: This Macro disables Group Modifiers processing. If the operation is successfully completed, all *GetDatum()* requests will again search for Datums in the Modifier List associated with the Current Device.

If an error was detected, a negative number (currently -1) is returned.

Errors are:

- No Current Device / Channel.
- Group Modifiers capability not available.

If the operation was successfully completed, a positive number (currently +1) is returned.

Note that, if Macro *SetGroupModsON()* returned a Good Status, so will this Macro. As currently implemented, there is no need to check `nStatus`.

See Also: *SetGroupModsON()*

Name: GetCurGroupModMOD

Type: Group Modifiers

Usage: `int nModCode;`
`nModCode = GetCurGroupModMOD();`

Description: This Macro returns the Modifier Code (**M_xxxx**) of the Current Group Modifier.

A negative number (currently -1) is returned if Group Modifiers processing is not enabled.

See Also: *SetGroupModsON()* and *SetGroupModsMOD()*

Name: GetCurGroupModLEVEL

Type: Group Modifiers

Usage: `int nGrpModLevel;
nGrpModLevel = GetCurGroupModLEVEL();`

Description: This Macro returns the Current Group Level.

A negative number (currently -1) is returned if Group Modifiers processing is not enabled.

See Also: *SetGroupModsON()*, *SetGroupModsMOD()* and *SetGroupModsLEVEL()*

Name: GetCurGroupModGROUP

Type: Group Modifiers

Usage: `int nGrpModGroup;
nGrpModGroup = GetCurGroupModGROUP();`

Description: This Macro returns the Current Group Number.

A negative number (currently -1) is returned if Group Modifiers processing is not enabled.

See Also: *SetGroupModsON()* and *SetGroupModsGROUP()*