

**Release Notes
RTS Version 3.9.21
12 June 2001**

1.0 Overview

This document describes changes included within:

<u>Module Name</u>	<u>Module Description</u>
SysConf	System Configuration
TPS	TYX Programming Support
PORF	PAWS Output Report Formatter
Targetter	ATLAS Program Targetter
XUtil	TYX X-Window Utilities
WinUtil	TYX MS Windows Utilities
PLI	PAWS LAPS Interpreter
PTE	PAWS Test Executive
CEM	PAWS CIIL Emulation Module

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

<u>Module</u>	<u>Changes</u>	<u>Rebuilt</u>	<u>Current Version</u>	
SysConf	Major	N/A	20010609	
TPS	Major	Yes	20010610	
PORF	None	Yes	20010612	3.5.4
Targetter	None	Yes	20010612	3.8.5
Xutil	None	Yes	20010612	
WinUtil	None	Yes	20010612	
PLI	Major	Yes	20010612	3.9.14
PTE	None	Yes	20010612	3.9.21
CEM	None	Yes	20010612	3.9.21

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

System Configuration - New System Identifier SYS12F - MSVC++ Version 6 (Windows NT only)

Prior to this release, all Modules distributed to Windows NT Customers were built under the System Configuration for SYS12B using the Microsoft Visual C++ Version 2 software development tools; System Configuration did not support the MSVC++ Version 6 software development tools.

With this release, System Configuration has a new System Identifier SYS12F for support of the MSVC++ Version 6 software development tools. All Modules have been built and tested under SYS12F. However, as of the date of this document, the only executable SYS12F Module that will be distributed to Windows NT Customers is PLI.

<u>Module(s)</u>	<u>System Configuration</u>
All except PLI	System 12B - MSVC++ Version 2.
PLI without Licensing	System 12B - MSVC++ Version 2.
PLI with Licensing	System 12F - MSVC++ Version 6.

Note The determination as to which PLI Module to distribute to a particular Windows NT Customer is the responsibility of the TYX Distribution Manager.

TYX Programming Support - New License Subroutines (LCS)

Prior to this release, TPS did not provide any support for licensing. Those executable Modules that were subject to licensing each contained their own licensing code.

With this release, TPS has a new LCS Library File (**lcs.lib**) and two new LCS Header Files (**LCdefs.h** and **LCfunc.h**) that support the details of License Client/Server processing with the following LCS Functions:

<u>Function</u>	<u>Description</u>
<i>LCinit()</i>	int LCinit(void)

Standard TPS Initialization Function that always returns zero.

Function Description

LCexit() **void LCexit(int nStatus)**

Program Termination Function that never returns. Must be called by Modules using LCS instead of calling TPS Operating System Subroutine (OSS) Function *OSexit()*. **nStatus** must be one of the Exit Status definitions (**OSSES???**) defined in OSS Header File **OSdefs.h**. Performs the following functions:

- Releases acquired License from License Server and/or terminates licensing processing.
- Calls TPS Log File (LFS) Function *LFclos()* to terminate Log File processing.
- Calls OSS Function *OSexit(nStatus)* to terminate program.

LCget() **int LCget(Argument(s))**

License Acquisition Function that returns **OSRSOK** if license acquired or **OSRSERR** if license not acquired. *Argument(s)* depends on the particular License Client/Server Model being used as defined by Define Labels **LCSTYPE?** in LCS Header File **LCdefs.h**:

<u>Model</u>	<u>Argument(s)</u>
LCSTYPE0	No licensing. LCget() not defined and must not be called.
LCSTYPE1	UNIX Licensing not yet implemented (with no plan to implement in the future).
LCSTYPE2	VENIX System V (SYS03I) not yet implemented (with no plan to implement in the future).
LCSTYPE3	SYS12F Licensing. Arguments: #1 = Long Integer containing Feature Identifier. Zero (0L) means no license required. May be over-ridden in Test Version of executable Program by setting Environment Variable TYXLCSFEATUREID to a Feature Identifier (including zero).

PAWS LAPS Interpreter - New License Support (Windows NT / SYS12F only)

Prior to this release, PLI was not subject to licensing for Customers using Windows NT.

With this release, PLI is subject to licensing for Customers using Windows NT. PLI uses the new TPS LCS Functions to perform the bulk of the License Client/Server processing. The determination as to which Customers will receive the SYS12F PLI is the responsibility of the TYX Distribution Manager.

PAWS LAPS Interpreter - New Support for Meta-Character Strings

Machine Language (ML) Function *scan()* supports Meta-Character Strings specified as:

\ **X X X** \

where: **XXX** is the two- or three-character Meta-Character String.

Prior to this release, Machine Language (ML) Function *scan()* did not correctly support Meta-Character Strings that contained Digit Characters.

With this release, ML Function *scan()* correctly supports Meta-Character Strings that contain Digit Characters. The following Meta-Character Strings are supported: **ACK, BEL, BS, CAN, CR, DC1, DC2, DC3, DC4, DEL, DLE, EM, ENQ, EOT, ESC, ETB, ETX, FF, FS, GS, HT, LF, NAK, NUL, RS, SI, SO, SOH, STX, SUB, SYN, US and VT.**

PAWS LAPS Interpreter - New Debug Capability For LAPS Source Code

Prior to this release, Debug Trace Functions *a_trc()* (Arithmetic Stack Trace) and *walkback()* (Call Stack Trace) were "static" Functions within PLI that were not available to LAPS Source Code.

With this release, new Machine Language (ML) Functions *ArithStkTrc()* (Arithmetic Stack Trace) and *CallStkTrc()* (Call Stack Trace) have been added to provide additional debug capability to LAPS Source Code:

ArithStkTrc() takes no Arguments. It prints the Character String "ArithStkTrc" and then calls *a_trc()* to "dump" the Arithmetic Stack. Nothing is returned to the Caller.

CallStkTrc() takes either no Arguments or one Character String Argument. (If no Arguments are specified, the default Character String "ML CallStkTrc" is used.) It calls *walkback()* with the specified Character String to "dump" the Call Stack. Nothing is returned to the Caller.

1.2 Problem Reports

None