Take Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page v.

First Edition (September 1994)

This edition to Version 3 Release 1 Modification 0, of IBM Application System/400 ILE C/400 (program 5763-CX2) and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the proper edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for readers’ comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Canada Ltd. Laboratory, Information Development
2G/345/1150/TOR
1150 Eglinton Avenue East
North York, Ontario, Canada. M3C 1H7

You can also send your comments by facsimile (attention: RCF Coordinator), or you can send your comments electronically to IBM. See “Communicating Your Comments to IBM” for a description of the methods. This page immediately precedes the Readers’ Comment Form at the back of this publication.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Notices

Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent product, program or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 208 Harbor Drive, Stanford, Connecticut, USA 06904-2501.

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Programming Interface Information

This document is intended to help you create Integrated Language Environment (ILE) C/400 programs. It contains information necessary to migrate your applications to the ILE C/400 compiler. It documents general-use programming interfaces and associated guidance information provided by the ILE C/400 compiler.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*), in this publication, are trademarks of the IBM Corporation in the United States or other countries:

Application System/400	Integrated Language Environment
AS/400	OS/400
C/400	PROFS
IBM	SAA
IBMLINK	Systems Application Architecture
ILE	400

Industry Standards

The ILE* C/400* compiler and library are designed according to the following standards:

- ANSI for C Programming Languages - C ANSI/ISO 9899-1990
- Systems Application Architecture* (SAA*) C level 2 interface as defined in *ILE C/400 Programmer's Reference*, SC09-1821.

About This Guide

This migration guide describes the changes you must make to your existing application if you currently use the Version 2 Release 3 (V2R3) System C/400 Programming RPQ P10102, and plan to migrate to the Version 3 Release 1 (V3R1) OS/400* system and Version 3 Release 1 ILE C/400.

This guide does not describe how to program in the C programming language, nor does it explain the concepts of ILE. See the *ILE Concepts*, SC41-3606, for more information about ILE.

See the following manuals for ILE C/400 specific information:

- *ILE C/400 Programmer's Guide*, SC09-1820
- *ILE C/400 Programmer's Reference*, SC09-1821
- *ILE C/400 Reference Summary*, SX09-1288

Who Should Use This Guide

This guide is for programmers who are familiar with the C programming language who currently have System C/400 applications and who plan to migrate those applications to ILE C/400 applications.

You need experience in using applicable AS/400 menus and displays or Control Language (CL) commands.

You also need knowledge of ILE as explained in the *ILE Concepts*.

Control Language Commands

If you need prompting, type the CL command and press F4 (Prompt). If you need online help information, press F1 (Help) on the CL command prompt display. See *CL Reference*, SC41-3722 for command syntax for the CL commands including ILE commands. CL commands can be used in either batch or interactive mode, or from a CL program.

Note: You need object authority to use CL commands. The *Security – Basic*, SC41-3301 contains information on object authority.

Migration to ILE C/400

If you choose to migrate your existing System C/400 programs to ILE C/400 programs, you must consider the following differences between the System C/400 Programming RPQ and the ILE C/400 product:

- A different way to access the PASA/PSSA header
- A different set of CL commands to create program objects
- No requirement to use setjmp and longjmp in signal handlers around MI function calls.
- Some machine interface instructions are not available in ILE C/400. They have been replaced by ILE C/400 equivalent machine interface instructions.

To aid in migrating code, you can run the Convert C Source (CVTCSRC) migration tool against your System C/400 source. This tool helps you locate required changes, and where possible, can make the changes for you. Changes are made for you only where there is a one-to-one typedef or macro name change.

You can also use the header file <tmcsysc.h> to help migrate code. This header file contains mappings from the System C/400 macro and typedef names to the ILE C/400 equivalent macro and typedef names. This helps you compile your code without changing macro and typedef names.

Note: Use the header file during the time you are migrating code, but do not forget to change the macro and typedef names in your source using the CVTCSRC tool.

The CVTCSRC tool and the header file <tmcsysc.h> are targeted to V2R3 System C/400 to V3R1 ILE C/400 migration. There are some specific V2R3 System C/400 to ILE C/400 changes that are not flagged by CVTCSRC or mapped by <tmcsysc.h>. These are indicated in this newsletter. Both the CVTCSRC migration tool and the <tmcsysc.h> header file are available in the QUSRTOOL library.

CL Commands

In the V3R1 ILE C/400 product, you use the following set of commands to create program objects:

- The CRTCMOD command compiles an ILE C/400 source member and produces a module object (type *MODULE). You must then use the CRTPGM command to create a program object (type *PGM) from one or more module objects.
- The CRTBNDC command creates a module object and then produces a program object (type *PGM) in one step. You can use this command instead of the CRTCMOD command followed by the CRTPGM command to create a program from a single module.

See the *ILE C/400 Programmer's Guide* for more information.

If any of your tools or applications refer to the V2R3 System C/400 commands you must change them to use the ILE C/400 commands to create objects.

Closing Files

In the V2R3 System C/400 Programming RPQ, files were not closed automatically when a program stopped running; in the V3R1 ILE C/400 product they are automatically closed.

Messages

In the V2R3 System C/400 Programming RPQ, the flag to turn off messages is PSE_MSG; in the V3R1 ILE C/400 product the flag is _C2M_MSG.

Signal Handling

If you are calling an ILE C/400 machine interface function and an exception occurs, the signal handler returns to the point of the call. Unlike System C/400, it is no longer necessary to use the setjmp and longjmp functions to ensure that the signal handler returns to the next statement following the function call. This was necessary since in System C/400, a machine interface function could expand to multiple machine interface instructions which may rely on the successful execution of the previous instruction. In ILE C/400, however, the runtime moves the resume point following an exception to the statement immediately following the function call.

Zoned and Packed Decimal Support

In System C/400, arithmetic comparison and copy operations on zoned and packed decimals were supported through the addn, cmpnv, cpynv, divn, mult, and subn machine interface library functions. Since the ILE C/400 compiler has a native packed decimal data type, only the cpynv operation is supported in ILE C/400. If you were using these operations for zoned decimal data, and since zoned decimal data is converted implicitly to packed decimal for arithmetic operations, it is recommended that where possible you should now perform these arithmetic operations in packed decimal. The cpynv function, _CPYNV builtin, _LBCPYNV builtin, the QXXZTOI (zoned to integer), QXXDTOZ (double to zoned), QXXZTOD (zoned to double) and QXXITOZ (integer to zoned) functions are available to convert to and from zoned and the native C data types.

Machine Interface Instructions

As a result of architecture changes for ILE (for example, the new storage model), some machine interface (MI) instructions have been replaced with ILE equivalents. While these instructions continue to be supported in native OS/400, they will not be available in ILE, having been completely replaced by the new instructions with equivalent (or enhanced) function. Porting System C/400 customers using these instructions, such as MATINVE, MATINVS, will have to change their code to use the ILE equivalents MATINVAT and FNDRINVN.

Some machine interface instructions, such as actpg, deactpg, and many exception handling MI instructions, do not apply to ILE programs. In these cases, no equivalents are provided. Where support for an machine interface instruction has been dropped in ILE (because it can be done as efficiently in "C") either a "C" version will be offered, or the equivalent "C" code maps directly to an existing library function or operation.

For those instructions that are replaced with a "C" function equivalent, although the interface has not changed from System C/400, there are some semantic differences. In most cases, this is limited to error situations such as expected exceptions. See the "Notes on Usage" for each MI function in the *ILE C/400 Programmer's Reference* for any differences in expected exceptions.

Table 1 lists System C/400 machine interface functions that are not supported in the ILE C/400 product. In most cases, there is an equivalent function. If you plan to migrate your program to the V3R1 ILE C/400 product, and your System C/400 program contains any of the functions listed under the column System C/400 Function, you must change your program as suggested in the column ILE C/400 Equivalent Function.

Table 1 (Page 1 of 2). System C/400 Migration

System C/400 Function	**ILE C/400 Equivalent Function**
actpg	Cannot be used to activate ILE programs. ILE programs are activated by program call instructions only.
addn	Use the ILE C/400 packed decimal arithmetic support
verify/Verify	Use the strspn library function
cmpnv	Use the ILE C/400 packed decimal data type
cvtcnd	Use the ILE C/400 packed decimal data type, cpynv, _CPYNV or _LBCPYNV
cvtcni	Use the ILE C/400 packed decimal data type, cpynv, _CPYNV or _LBCPYNV
cvtncd	Use the ILE C/400 packed decimal data type, cpynv, _CPYNV or _LBCPYNV
cvtnci	Use the ILE C/400 packed decimal data type, cpynv, _CPYNV or _LBCPYNV
deactpg	Cannot be used to deactivate ILE programs. ILE program activations are destroyed when the containing activation group is destroyed.
div	Use the ILE C/400 packed decimal arithmetic support
matinat	Not applicable in ILE
matinv	Not applicable in ILE
matinve	Use matinvat, _MATINVAT1, or _MATINVAT2 and fndrinvn, _FNDRINVN1, or _FNDRINVN2
matinvs	Use matinvat, _MATINVAT1, or _MATINVAT2 and fndrinvn, _FNDRINVN1, or _FNDRINVN2
Matqmsg	Use matqmsg or _MATQMSG
mult	Use the ILE C/400 packed decimal arithmetic support
retexcpd	Use the Message Handler API QMHRCVPM
Rslvsp	Use rslvsp or _RSLVSPn (n = 1 to 8 incl.)
rtnexcp/Rtnexcp	Not applicable in ILE

Table 1 (Page 2 of 2). System C/400 Migration	
System C/400 Function	**ILE C/400 Equivalent Function**
sigexcp/Sigexcp	Use the Message Handler API QMHSNDPM
sigexcpr/Sigexcpr	Use the MH API QMHRSNEM
snsexcpd/Snsexcpd	Not applicable in ILE
ssca	Use retca and setca
subn	Use the ILE C/400 packed decimal arithmetic support
Scanwc	Use scanwc
Testau	Use testau, _TESTAU1 or _TESTAU2
Triml	Use triml
Xlatewt	Use xlatewt or _XLATEB

Header files

Machine interface header files are prefixed with *mi*. All macro names in these header files are in upper case to more closely follow C naming conventions for #define names. The following is a list of the machine interface header files provided with the ILE C/400 library:

miauth.h
milib.h
mimchobs.h
miproces.h
mirsc.h
mitime.h
micomput.h
milock.h
mipgexec.h
miptrnam.h
mispace.h
miindex.h
mimchint.h
mipgmgmt.h
miqueue.h
mispcobj.h

See the *ILE C/400 Programmer's Reference* for a complete list and description of the machine interface instructions supported in the V2R3 ILE C/400 product.

There are a number of changes to these header files for ILE C/400. Changes are listed under each header file.

<miauth.h>

1. In the V3R1 ILE C/400 product, the macro Testau is not supported. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses this macro, you must now use one of testau, _TESTAU1, or _TESTAU2.
2. <tmcsysc.h> contains mappings for the following System C/400 macro names:

```
_AUTH_Obj_Ctrl  ->  _AUTH_OBJ_CTRL
_AUTH_Obj_Mgmt  ->  _AUTH_OBJ_MGMT
_AUTH_Pointer   ->  _AUTH_POINTER
_AUTH_Space     ->  _AUTH_SPACE
_AUTH_Retrieve  ->  _AUTH_RETRIEVE
_AUTH_Insert    ->  _AUTH_INSERT
_AUTH_Delete    ->  _AUTH_DELETE
_AUTH_Update    ->  _AUTH_UPDATE
_AUTH_Owner     ->  _AUTH_OWNER
_AUTH_Excluded  ->  _AUTH_EXCLUDED
_AUTH_Lst_Mgmt  ->  _AUTH_LST_MGMT
_AUTH_Any       ->  _AUTH_ANY
```

<micomput.h>

1. The following functions are not supported in the V3R1 ILE C/400 product. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses these functions, refer to Table 1 on page 3 for the ILE C/400 equivalents.

addn	cmpnv	cvtcnd	cvtcni
cvtncd	cvtnci	div	mult
ssca	subn	verify	

2. The following macros are not supported in the V3R1 ILE C/400 product. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses these macros, refer to Table 1 on page 3 for the ILE C/400 equivalents.

   - Scanwc
   - Triml
   - Verify
   - Xlatewt

   In the V3R1 ILE C/400 product, "OR-ing" the values for the options operand on scanwc is supported. This was not permitted for System C/400. In System C/400, if the constants provided for the options were "OR-ed" together, (for example, _SCWC_mixed_less|_SCWC_mixed_equal), the result returned by scanwc was always "-1" (not found).

3. The following typedefs associated with ssca are not supported in the V3R1 ILE C/400 product.

   - typedef struct _SSCA_Float_T
   - typedef struct _SSCA_Template_T

   In the V3R1 ILE C/400 product, retca and setca are used to return and store the computational attributes. The System C/400 ssca function is not available in ILE C/400. If the System C/400 program you are migrating to ILE C/400 uses the ssca function, you will need to change your code to use retca or setca, depending on your usage of ssca. Although the interface to return or set the computational attributes is only slightly different, the semantic difference is significant.

   Computational attributes are saved and restored at the program entry procedure boundary only. At that time they are used to set the C language defaults. Computational attributes are not saved and restored at any other point. For example, ssca will save and restore computational attributes across all program call boundaries. However, computational attributes changed through setca will remain in effect until changes through a subsequent setca or until the original attribute settings are restored when the program entry procedure returns.

4. The Conversion_stt[2] field in the _CVTSC_Control_T template was changed to a short. This was done to make accessing the field easier. In System C/400, the cpyblap function converts negative lengths to large positive values. For V3R1 ILE C/400, the negative length has the same result as zero.

5. <tmcsysc.h> contains mappings for the following System C/400 macro names:

```
_CVTCM_compress              -> _CVTCM_COMPRESS
_CVTCM_truncate              -> _CVTCM_TRUNCATE
                             ->
_CVTMC_Move_SRCB             -> _CVTMC_MOVE_SRCB

_CVTCS_no_compression        -> _CVTCS_NO_COMPRESS
_CVTCS_compress              -> _CVTCS_COMPRESS
_CVTCS_rec_sep_no_con        -> _CVTCS_REC_SEP_NO_CON
_CVTCS_rec_sep_con           -> _CVTCS_REC_SEP_CON
_CVTCS_record_span           -> _CVTCS_RECORD_SPAN

_CVTSC_no_decompression      -> _CVTSC_NO_DECOMPRESS
_CVTSC_decompress            -> _CVTSC_DECOMPRESS
_CVTSC_rec_sep_no_con        -> _CVTSC_REC_SEP_NO_CON
_CVTSC_rec_sep_con           -> _CVTSC_REC_SEP_CON
_CVTSC_mov_rec_sep           -> _CVTSC_MOV_REC_SEP
_CVTSC_trans_rec_sep         -> _CVTSC_TRANS_REC_SEP
_CVTSC_mv_rec_sep_con_to_rec -> _CVTSC_MOV_REC_SEP_CON_TO_REC

_CVTSC_convert_no_trans      -> _CVTSC_CONVERT_NO_TRANS
_CVTSC_convert_trans         -> _CVTSC_CONVERT_TRANS
                             ->
_SCWC_nonmixed_equal         -> _SCWC_NONMIXED_EQUAL
_SCWC_nonmixed_less          -> _SCWC_NONMIXED_LESS
_SCWC_nonmixed_greater       -> _SCWC_NONMIXED_GREATER
_SCWC_mixed_equal            -> _SCWC_MIXED_EQUAL
_SCWC_mixed_less             -> _SCWC_MIXED_LESS
_SCWC_mixed_greater          -> _SCWC_MIXED_GREATER
```

<miexcept.h>

1. The header file <miexcept.h> is not supported in the V3R1 ILE C/400 product. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses any of the following functions, macros, or typedefs, refer to Table 1 on page 3 for the ILE C/400 equivalents.

   - Functions

     ```
     retexcpd
     rtnexcp or Rtnexcp
     sigexcp or Sigexcp
     sigexcpr or Sigexcpr
     snsexcpd or Snsexcpd
     ```

   - Macros

     ```
     #define _REXD_Branch   0x00
     #define _REXD_Internal 0x01
     #define _REXD_External 0x02

     #define _REX_Resume    0x01
     #define _REX_Indirect  0x02

     #define _SE_Normal   0x00
     #define _SE_No_PDEH  0x40
     #define _SE_Descript 0x20
     ```

   - Typedefs

```
typedef _Packed struct _REXD_Template_T
typedef _Packed struct _SEXD_Template_T
typedef _Packed struct _SEAT_Template_T
typedef _Packed struct _SEED_Template_T
```

<miindex.h>

1. <tmcsysc.h> contains mappings for the following System C/400 macro names:

```
_Equals         ->  _FIND_EQUALS
_Greater        ->  _FIND_GREATER
_Lesser         ->  _FIND_LESSER
_Not_Lesser     ->  _FIND_NOT_LESSER
_Not_Greater    ->  _FIND_NOT_GREATER
_First          ->  _FIND_FIRST
_Last           ->  _FIND_LAST
_Between        ->  _FIND_BETWEEN

_Insert         ->  _INSERT
_Insert_Repl    ->  _INSERT_REPLACE
_Insert_No_Repl ->  _INSERT_NO_REPLACE
```

<milib.h>

1. The macro USER_RETURN_CODE is not supported in the V3R1 ILE C/400 product. Its location in the Languages and Utilities work area of the WCB has been replaced with the ILE language return code, represented by the _LANGUAGE_RETURN_CODE macro.
2. In the V2R3 System C/400 Programming RPQ, the _LANGUAGE_RETURN_CODE macro returns the two byte Language and Utilities return code. In the ILE C/400 product, the _LANGUAGE_RETURN_CODE macro returns the four byte ILE language return code. All ILE programs set this four byte return code. Only OPM programs will still set the two byte Languages and Utilities return code.
3. In the V2R3 System C/400 Programming RPQ, the typedefs for _OBJ_NAME and _LIB_NAME appear in the header file <milib.h>. In the ILE C/400 product, the typedefs for _OBJ_NAME and _LIB_NAME appear in the header file <miptrnam.h> where they are referenced.
4. In the V2R3 System C/400 Programming RPQ, the macro NUM_DESCR macro and the typedef _NUM_Descr_T appear in the header file <milib.h>. In the ILE C/400 product, the macro NUM_DESCR macro and the typedef _NUM_Descr_T appear in the header file <micomput.h> where they are referenced.
5. In the V2R3 System C/400 Programming RPQ, the typedef _DPA_Template_T appears in the header file <milib.h>. In the ILE C/400 product, the typedef _DPA_Template_T appears in the header file <micomput.h> where it is referenced.
6. In the V2R3 System C/400 Programming RPQ, the following macros appear in the header file <milib.h>. In the ILE C/400 product, these macros appear in the header file <micomput.h> where they are referenced. <tmcsysc.h> contains mappings for these System C/400 macro names:

```
T_Signed   ->    _T_SIGNED
T_Float    ->    _T_FLOAT
T_Zoned    ->    _T_ZONED
T_Packed   ->    _T_PACKED
T_Char     ->    _T_CHAR
T_Unsigned ->    _T_UNSIGNED
```

7. The following macro names have been removed:

```
T_Onlyns
T_Onlyn
T_Either
T_Open
```

8. _Access_Group has been removed from the _OBJ_TYPE_T enumeration data type. Access groups are system domain objects.
9. <tmcsysc.h> contains mappings for the following System C/400 macro names:

```
_AUTH_OBJCTL  -> _AUTH_OBJ_CTRL
_AUTH_OBJMGT  -> _AUTH_OBJ_MGMT
_AUTH_AUTPTR  -> _AUTH_POINTER
_AUTH_SPCAUT  -> _AUTH_SPACE
```

<milock.h>

1. <tmcsysc.h> contains mappings for the following System C/400 macro names:

```
_LSRD_Lock               -> _LSRD_LOCK
_LSRO_Lock               -> _LSRO_LOCK
_LSUP_Lock               -> _LSUP_LOCK
_LEAR_Lock               -> _LEAR_LOCK
_LENR_Lock               -> _LENR_LOCK
_LOCK_Entry_Active       -> _LOCK_ENTRY_ACTIVE

_LOCK_Asynch             -> _LOCK_ASYNCH
_LOCK_Synch              -> _LOCK_SYNCH
_LOCK_Wait_Enter_Mod     -> _LOCK_WAIT_ENTER_MOD
_LOCK_Wait_Leave_Mod     -> _LOCK_WAIT_LEAVE_MOD
_LOCK_Wait_Indefinite    -> _LOCK_WAIT_INDEFINITE
_LOCK_Normal             -> _LOCK_NORMAL

_UNLOCK_Specific         -> _UNLOCK_SPECIFIC
_UNLOCK_Held_or_Waiting  -> _UNLOCK_HELD_OR_WAITING
_UNLOCK_Waiting          -> _UNLOCK_WAITING

_UNLOCK_All              -> _UNLOCK_ALL
```

<mimchint.h>

1. In the V3R1 ILE C/400 product, the _MMTR_Template_T structure has changed and _MMTK2_Template_T has been removed. A substructure called Options has been added to _MMTR_Template_T and unioned with _MMTR_108. If your program refers to the typedef _MMTR2_Template_T or _MMTR_Template_T, you must change your program.
2. In the V3R1 ILE C/400 product, the matmatr2 function is not supported. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses this function, you must change your program to use either matmatr or _MATMATR1.

This change and the previous change were made to provide an interface to MATMATR through a single matmatr function.

3. In the V3R1 ILE C/400 product, the _MMTR_0104 template does not return the Process Communication Object Pointer (Comm_I). This is a reserved field.

   This change and the previous change result from the ILE architecture changes affecting PASA/PSSA usage.

4. In the V2R3 System C/400 Programming RPQ, the member names in the following templates are in lowercase: _MEM_VPD, _PROC_VPD, _COL_VPD, _CEC__VPD, _PANEL_VPD, and _MMTR_012C. In the V3R1 ILE C/400 product, the member names have the following consistent naming convention:

   - _MEM_VPD

     ```
     mem_status       ->  Mem_Status
     mem_card_size    ->  Mem_Card_Size
     CCINS_array      ->  CCINS_Array
     slot             ->  Slot
     ```

   - _PROC_VPD

     ```
     status           ->  Status
     model            ->  Model
     part             ->  Part
     serial           ->  Serial
     manufacture      ->  Manufacture
     load             ->  Load
     ```

   - _COL_VPD

     ```
     card_status      ->  Card_Status
     model            ->  Model
     part             ->  Part
     serial           ->  Serial
     manufacturing    ->  Manufacturing
     ```

   - CEC_VPD

     ```
     CEC_read         ->  CEC_Read
     manufacturing    ->  Manufacturing
     serial           ->  Serial
     type             ->  Type
     model            ->  Model
     pseudo_model     ->  Pseudo_Model
     ```

   - _PANEL_VPD

     ```
     panel_status     ->  Panel_Status
     panel_type       ->  Panel_Type
     model            ->  Model
     part             ->  Part
     serial           ->  Serial
     manufacturing    ->  Manufacturing
     ROS_part         ->  ROS_Part
     ROS_card         ->  ROS_Card
     ROS_flag         ->  ROS_Flag
     ROS_fix          ->  ROS_Fix
     ```

   - _MMTR_012C

```
Mem_installed    ->  Mem_Installed
Mem_required     ->  Mem_Required
mem_array        ->  Mem_Array
proc_array       ->  Proc_Array
col_array        ->  Col_Array
cec_info         ->  CEC_Info
panel_Info       ->  Panel_Info
```

5. <tmcsysc.h> contains mappings for the following System C/400 macro names:

```
_MMTR_Serial      ->    _MMTR_SERIAL
_MMTR_UPSDelay    ->    _MMTR_UPS_DELAY
_MMTR_DateForm    ->    _MMTR_DATE_FORMAT
_MMTR_LeapYear    ->    _MMTR_LEAP_YEAR
_MMTR_TPowerOn    ->    _MMTR_TIMED_POWER_ON
_MMTR_TPowerOnED  ->    _MMTR_TIMED_POWER_ON_ED
_MMTR_RPowerOnED  ->    _MMTR_REMOTE_POWER_ON_ED
_MMTR_APowerRED   ->    _MMTR_AUTO_POWER_RESTART_ED
_MMTR_DateSep     ->    _MMTR_DATE_SEP
_MMTR_UPS_Type    ->    _MMTR_UPS_TYPE
_MMTR_PanelStatus ->    _MMTR_PANEL_STATUS
```

```
typedef  _MMTR_0104_T     _MMTR_0104
typedef  _MMTR_ROL_C_T    _MMTR_ROL_C
typedef  _MMTR_ROL_D_T    _MMTR_ROL_D
typedef  _MMTR_ROL_I_T    _MMTR_ROL_I
typedef  _MMTR_ROL_P_T    _MMTR_ROL_P
typedef  _MMTR_ROL_J_T    _MMTR_ROL_J
typedef  _MMTR_0108_T     _MMTR_0108
typedef  _MMTR_0118_T     _MMTR_0118
typedef  _MEM_VPD_T       _MEM_VPD
typedef  _PROC_VPD_T      _PROC_VPD
typedef  _COL_VPD_T       _COL_VPD
typedef  _CEC_VPD_T       _CEC_VPD
typedef  _PANEL_VPD_T     _PANEL_VPD
typedef  _MMTR_012C_T     _MMTR_012C
typedef  _MMTR_0130_T     _MMTR_0130
typedef  _MMTR_013C_T     _MMTR_013C
typedef  _MMTR_0164_T     _MMTR_0164
typedef  _MMTR_0168_T     _MMTR_0168
typedef  _MMTR_016C_T     _MMTR_016C
```

<mimchobs.h>

1. In the V3R1 ILE C/400 product, the member names for the _MSOB_Template_T template have been changed as follows:

```
Obj_Dmn_Atr     ->   State_Given
                     reserved4
                     Pgm_Type

Init_Domain     ->   Pgm_State

Obj_Sz_in_page  ->   Size_In_Pages
```

2. In the V3R1 ILE C/400 product, the following function names have been changed as follows:

```
fndrinvn1  ->  _FNDRINVN1
fndrinvn2  ->  _FNDRINVN2
matinvat1  ->  _MATINVAT1
matinvat2  ->  _MATINVAT2
```

If you are using the function version of FNDRINVN or MATINVAT, there is a stack frame entry associated with it; however, if you are using the macro version there is not. Since the interface used affects the creation of a stack frame entry, this also has an effect on the relative invocation number returned by fndrinvn and used with matinvat.

In ILE C/400, these interfaces are available as builtins only. See matinvat and fndrinvn functions in the *ILE C/400 Programmer's Reference* for more information.

3. In the V3R1 ILE C/400 product, the matinat function is not supported. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses this function or any of the following macros and typedefs, they must be removed.

```
_MINA_Binary
_MINA_Float
_MINA_Zoned
_MINA_Packed
_MINA_Char
_MINA_Pointer

typedef _Packed struct _MINA_Data_Object_T
typedef _Packed struct _MINA_Data_Const_T
typedef _Packed struct _MINA_Instruction_T
typedef _Packed struct _MINA_Arg_List_T
typedef _Packed struct _MINA_Exception_T
typedef _Packed struct _MINA_Pointer_T
typedef _Packed struct _MINA_Template_T
```

4. In the V3R1 ILE C/400 product, the matinv function is not supported. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses this function, refer to Table 1 on page 3 for the ILE C/400 equivalents.
5. In the V3R1 ILE C/400 product, the matinve function is not supported. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses this function, refer to Table 1 on page 3 for the ILE C/400 equivalents. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses the following macros and typedefs, they must be removed:

```
_MIVE_Long
_MIVE_Short_1
_MIVE_Short_2
_MIVE_Short_3
_MIVE_Short_4
_MIVE_Short_5

_MIVE_Omit
_MIVE_Calle
_MIVE_Transfer
_MIVE_Event
_MIVE_Exception
_MIVE_Prob_St
_MIVE_Init_St
_MIVE_Term_St
_MIVE_Exit
typedef _Packed struct _MIVE_Long_T
typedef _Packed struct _MIVE_Short5_T
typedef struct _MIVE_Template_T
```

Please note, _MIVE_Short_5 and typedef _Packed struct _MIVE_Short5_T are not in <tmcsysc.h> because they are only available in the V2R3 System C/400 product.

6. In the V3R1 ILE C/400 product, the matinvs function is not supported. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses this function, refer to Table 1 on page 3 for the ILE C/400 equivalents. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses the following macros and typedefs, they must be removed:

   ```
   typedef struct _MIVS_Entry_T
   typedef struct _MIVS_Template_T
   ```

7. In the V2R3 System C/400 Programming RPQ, the pointer types in the list below appeared in the header file <mimchobs.h>. In the V3R1 ILE C/400 product, these pointer type macros have been renamed and placed in <milib.h>. <tmcsysc.h> contains mappings for the System C/400 macros __SYSPTR and __SPCPTR.

   ```
   __SYSPTR    ->   _PTR_T_SYS
   __SPCPTR    ->   _PTR_T_SPC

   __INVPTR    ->   _PTR_T_INV
   ```

8. The following macros have been removed since data pointers and instruction pointers are not used in the V3R1 C/400 ILE product:

   ```
   __DTAPTR
   __INSPTR
   ```

9. In V3R1 ILE C/400 the _MPTR_Template_T type is a union of four pointer templates. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses structures of this type, you must change your program.

10. In the V3R1 ILE C/400 product, the following macros have been removed:

```
#define SYP     "SYP:"
#define SPC     "SPP:"
#define DTA     "DTP:"
#define INS     "INP:"
#define NUL     "NULL"

#define __BIN     0x00
#define __FLT     0x01
#define __ZON     0x02
#define __PAC     0x03
#define __CHR     0x04
```

<mipgexec.h>

1. In the V3R1 ILE C/400 product, the following functions are not supported.

   ```
   actpg
   deactpg
   ```

2. In the V3R1 ILE C/400 product, the modasa function does not support automatic storage frame truncation. The value specified for the modification size must be greater than zero.

3. In the V3R1 ILE C/400 product, the semantics for the atiexit (At Invocation Exit) function map closely to System C/400. Like System C/400, the ILE C/400 atiexit function accepts a pointer to an OS-linkage function pointer only. However, because program pointers and procedure pointers are different data types in ILE C/400, it was necessary to make some changes to the atiexit function interface to accept an OS-linkage function pointer (i.e., a pointer to a program). The following illustrates the different interfaces.

   ```
   System C/400

   int atiexit(void (*func)(void*), void*);



   ILE C/400

         typedef void(_OS_func_t)(void*);
         #pragma linkage(_OS_func_t, OS)

         int atiexit(_OS_func_t *, void*);
   ```

   If the System C/400 program you are migrating to the ILE C/400 product uses the atiexit function, change the data type of the invocation exit handler to _OS_func_t *. Also, read 'Establishing an Invocation Exit Program' in the 'Machine Interface Library Functions' chapter of the *ILE C/400 Programmer's Reference* for details on how the atiexit semantics are mapped to ILE.

   To use a bound function rather than an OS-linkage function to implement an invocation exit handler, use the ILE APIs CEERTX, CEEUTX or a #pragma cancel handler. The semantics for the APIs are different from the atiexit MI function. See the *System API Programming*, SC41-3800 manual for more information on the CEERTX and CEEUTX APIs and the *ILE C/400 Programmer's Reference* for more information about cancel handlers.

<mipgmgmt.h>

1. In the V3R1 ILE C/400 product, the member name in the _MATPG_Template_T template has changed:

```
T_Extned  ->   T_Extend
```

<miproces.h>

1. In the V3R1 ILE C/400 product, option 0x15 does not return a Process Communication Object Pointer in the template. This means that Proc_Comm in the _MPRT_CI_T template is a reserved field.
2. In the V3R1 ILE C/400 product, the functions matpratr1, matpratr2, and matpratr3 are not supported. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses these functions, you must now use one of matpratr, _MATPRATR1, or _MATPRATR2.
3. In the V3R1 ILE C/400 product, the templates _MPRT_SP_T, _MPRT_PTR__T, and _MPRT_AU_T have been integrated into the _MPRT_Template_T template. As a result, _MPRT_Template_T has become a union with the substructure Scalar_Attr added. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses any of these templates, it must be changed.
4. <tmcsysc.h> contains mappings for the following System C/400 macro names:

```
_MPRT_All          ->   _MPRT_ALL
_MPRT_Proc_Ctrl    ->   _MPRT_PROC_CTRL
_MPRT_Ctrl_Mask    ->   _MPRT_CTRL_MASK
_MPRT_Num_Monit    ->   _MPRT_NUM_MONIT
_MPRT_Proc_Pri     ->   _MPRT_PROC_PRTY
_MPRT_Stor_Pool    ->   _MPRT_STOR_POOL
_MPRT_Max_Stor     ->   _MPRT_MAX_STOR
_MPRT_Time_Int     ->   _MPRT_TIME_INT
_MPRT_Time_Out     ->   _MPRT_TIME_OUT
_MPRT_Max_Time     ->   _MPRT_MAX_TIME
_MPRT_Proc_Clas    ->   _MPRT_PROC_CLASS
_MPRT Proc_Cat     ->   _MPRT_PROC_CAT
_MPRT_Ctrl_Ind     ->   _MPRT_CTRL_IND
_MPRT_User_Prof    ->   _MPRT_USER_PROF
_MPRT_Proc_Comm    ->   _MPRT_PROC_COMM
_MPRT_Proc_Name    ->   _MPRT_PROC_NAME
_MPRT_Init_Pgm     ->   _MPRT_INIT_PGM
_MPRT_Term_Pgm     ->   _MPRT_TERM_PGM
_MPRT_Prob_Pgm     ->   _MPRT_PROB_PGM
_MPRT_Except_H     ->   _MPRT_EXCEPT_H
_MPRT_Stat_Ind     ->   _MPRT_STAT_IND
_MPRT_Rsc_Usage    ->   _MPRT_RSC_USAGE
_MPRT_Sub_Proc     ->   _MPRT_SUB_PROC
_MPRT_Proc_Perf    ->   _MPRT_PROC_PERF
_MPRT_Exec_Stat    ->   _MPRT_EXEC_STAT
_MPRT_Ctrl_Spc     ->   _MPRT_CTRL_SPC
_MPRT_Adopt_UP     ->   _MPRT_ADOPT_UP

_Wait_Normal       ->  _WAIT_NORMAL
_Wait_Mod_AS_Enter ->  _WAIT_MOD_AS_ENTER
_Wait_Mod_AS_Leave ->  _WAIT_MOD_AS_LEAVE
_Wait_MPL_Set      ->  _WAIT_MPL_SET
```

<miptrnam.h>

1. In the V2R3 System C/400 Programming RPQ, the pointer type macros in the list below appeared in the header file <miptrnam.h>. In the V3R1 ILE C/400 product, these pointer type macros have been renamed and placed in <milib.h>. <tmcsysc.h> contains mappings for the System C/400 macro names:

   ```
   NULLPTRCMP    ->   _PTR_T_NULL
   SYSPTRCMP     ->   _PTR_T_SYS
   SPCPTRCMP     ->   _PTR_T_SPC
   ```

2. The following macros have been removed since data pointers and instruction pointers are not used in the V3R1 ILE C/400 product.

   ```
   DTAPTRCMP
   INSPTRCMP
   ```

3. In the V3R1 ILE C/400 product, the macro Rslvsp is not supported. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses this macro, you must use with rslvsp or _RSLVSPn.

<miqueue.h>

1. In the V3R1 ILE C/400 product, the member names for the template _ENQ_Msg_Prefix_T have been changed as follows:

   ```
   msg_len -> Msg_Len
   MSG     -> Msg
   ```

2. In the V3R1 ILE C/400 product, the member names for the template _DEQ_Msg_Prefix_T have changed as follows:

   ```
   wait_time    -> Wait_Time
   time_stamp   -> Time_Stamp
   msg_size     -> Msg_Size
   acc_state1   -> Acc_State1
   acc_state2   -> Acc_State2
   mpl          -> MPL
   wait_forever -> Wait_Forever
   ```

3. In the V3R1 ILE C/400 product, the macro Matqmsg is not supported. If the System C/400 program you are migrating to the V3R1 ILE C/400 product uses this macro, you must now use either matqmsg or _MATQMSG.

4. <tmcsysc.h> contains mappings for the System C/400 macro names and typedefs:

   ```
   typedef  _DEQ_Msg_Prefix_T  _DEQ_msg_prefix;

   typedef  _ENQ_Msg_Prefix_T  _ENQ_msg_prefix;

   _MQMS_All         ->  _MQMS_ALL
   _MQMS_First       ->  _MQMS_FIRST
   _MQMS_Last        ->  _MQMS_LAST
   _MQMS_Keyed       ->  _MQMS_KEYED

   _MQMS_Greater     ->  _MQMS_GREATER
   _MQMS_Lesser      ->  _MQMS_LESSER
   _MQMS_Not_Equal   ->  _MQMS_NOT_EQUAL
   _MQMS_Equal       ->  _MQMS_EQUAL
   _MQMS_Not_Lesser  ->  _MQMS_NOT_LESSER
   _MQMS_Not_Greater -> _MQMS_NOT_GREATER
   ```

<mirsc.h>

<tmcsysc.h> contains mappings for the System C/400 macro names:

```
_SETACST_ASYNCH_ASSUR      -> _SETACST_ASYNCH_ASSURE

_MATRMD_Process_Data       -> _MATRMD_PROCESS_DATA
_MATRMD_Storage_Counters   -> _MATRMD_STORAGE_COUNTERS
_MATRMD_Transient_Pool     -> _MATRMD_TRANSIENT_POOL
_MATRMD_Address_Threshold  -> _MATRMD_ADDRESS_THRESHOLD
_MATRMD_Main_Pool          -> _MATRMD_MAIN_POOL
_MATRMD_MPL_Control        -> _MATRMD_MPL_CONTROL
_MATRMD_Reserved_Pool      -> _MATRMD_RESERVED_POOL
_MATRMD_User_Storage       -> _MATRMD_USER_STORAGE
_MATRMD_Aux_Storage        -> _MATRMD_AUX_STORAGE
_MATRMD_Intervals          -> _MATRMD_INTERVALS
```

In the V3R1 ILE C/400 product, the _MATRMD_0x13_Template_T template has been changed. The template no longer contains the nested structure _MATRMD13_T. If the System C/400 product you are migrating to the ILE C/400 product uses structures of this type, you must change your program.

<mispace.h>

1. In the V3R1 ILE C/400 product, the following field names in _SPC_Template_T have been replaced with reserved fields:

   ```
   Public
   Owner
   Set_Public
   RecoveryOpts
   ```

Pointers

1. If the System C/400 programs that you are migrating to ILE C/400 programs contain the machine interface instruction matinve you must change the program to use the matinvat instruction. The following example illustrates how to change your source program. This example also shows how the program pointer for the current invocation can be retrieved in System C/400 and in ILE C/400 using the matinvat function.

System C/400 version

```
#include <mimchobs.h>

void func(void) {
  _SYSPTR            pgm_ptr;
  _MINVE_Template_T  matinve_t;
      :
  matinve(matinve_t, _MINVE_Short_1);
  pgm_ptr = matinve_t.Data_1;
      :
}
```

ILE C/400 version

```
#include <pointer.h>
#include <mimchobs.h>

void func(void) {
  _SYSPTR   pgm_ptr;
      :
  matinvat(&pgm_ptr, NULL, _MTVA_PGM_PTR,
           sizeof(pgm_ptr))
      :
}
```

Index