ISOGON CORPORATION

INITIAL PROGRAMMING FUNCTIONAL SPECIFICATIONS

INTERCOMM SVC INTEGRITY SUPPORT ENHANCEMENT

Document Date: October 17, 1989

ISOGON CONFIDENTIAL

330 Seventh Avenue • New York, New York 10001 • (212) 967-2424 • Telex 888302 • FAX (212) 967-3198

1. GENERAL OBJECTIVES

1.1 PROGRAMMING SYSTEM

This support will become part of Intercomm Release 10.

1.2 SUMMARY OF SPECIFICATIONS

The new Intercomm SVC support provides for an environment in which no part of Intercomm (except the SVC routine) ever executes in anything but problem state and problem key, as an unauthorized program.

All functions that currently require that Intercomm executes in system key and/or supervisor state will be moved into the Intercomm SVC routine, and that routine will provide full integrity support.

1.3 SUMMARY OF MIGRATION AND COEXISTENCE CHARACTERISTICS

The new support will be part of Intercomm Release 10, and will only run on a system running MVS/SP2 (MVS/XA) or MVS/SP3 (MVS/ESA).

Installations running Intercomm Release 9 may continue to do so, but will not be able to utilize the enhanced integrity control provided by this new support.

Installations running Intercomm Release 10 on a system with MVS/SP1 (MVS/370) may continue to do so, but will then not be able to utilize the enhanced integrity control provided by this new support.

The new support will be shipped as a System Modification (SM), and will be supported in a way similar to the way Intercomm is currently supported.

Installations that prefer not to install this SM may continue to use the existing SVC routine, but Isogon Corporation will only accept integrity-related Maintenance Service Requests (MSRs) that are reproducible on a system with this SM applied.

2. USER REQUIREMENTS

The existing Intercomm SVC routine is a necessary part of Intercomm in support of above all the Multi-Region Support (MRS) facility and the Extended Security System (ESS) facility. The SVC was written many years ago, when the general security and integrity awareness was much lower than it is today.

The SVC was created to provide, to the Intercomm program, the required basic ability to switch into and out of supervisor state and/or system key. In addition, it contains several functional parts of Intercomm, such as POSTing an ECB located in the link-pack area.

When the SVC routine was created, the need for high-performance paths through all system software was high, and often questions such as security and integrity were given a back seat, as long as performance was good.

In today's environment, with ever faster computers and much more concern for system and data integrity, the simplistic approach taken when the Intercomm SVC was created is no longer acceptable. Today, system integrity must take precedence over issues such as performance (even though performance is still very important in a product such as Intercomm).

Therefore, the Intercomm SVC routine must be rewritten so that it adheres to the system integrity requirements, as defined by IBM.

It should be noted, that **system integrity** is not concerned with issues such as an unauthorized user's ability to affect overall system performance and availability through excessive use of resources such as CPU time, System Queue Area (SQA) space, etc., nor is it concerned with **system security**, except to the extent that it must not be possible to bypass the security checks in the system through the unauthorized use of a system facility.

Thus, while the existing SVC routine will be modified so that all of Intercomm conforms with the above integrity requirements, Intercomm will continue to rely upon other facilities, such as RACF or ACF2 for data security, the Intercomm ESS facility for password validation and access protection, and normal installation controls to prevent "non-trusted" user code from being installed as an Intercomm subsystem or user exit.

When the changes described in this document have been implemented, it is Isogon Corporation's intent to provide a corporate integrity statement relating to Intercomm, similar to IBM's integrity statement about MVS, stating that any integrity problem found in Intercomm, as distributed by Isogon Corporation, will be corrected, and that no valid integrity MSR can be "corrected" via a change in publications.

3. FUNCTIONAL CHARACTERISTICS

3.1 DESCRIPTION

3.1.1 EXISTING SVC FUNCTIONS

The Intercomm SVC currently provides a wide variety of system services to the Intercomm product. The functions are as follows:

- Set protection key to 0
- Reset protection key
- Set supervisor state
- Set problem state
- Terminate a task (not used)
- Clear a protected-memory ECB
- Clear a protected-memory ECB if the wait flag is on (not used)
- Clear a protected-memory ECB if the post flag is on
- Post a protected-memory ECB
- Post an ECB, using cross-memory post

These services are used by various Intercomm system modules, especially startup, MRS, and ESS. The first four functions simply change the key or state of the caller (Intercomm), while the others perform a specific function and return control in the caller's original key and state.

These functions perform no, or very limited, validation to ensure that the caller is authorized to use the function, and that the data areas referenced belong to the caller.

Outside of the SVC, some of the Intercomm functions that are performed in either key 0 or supervisor state (or both) are:

- Cross-Memory wait on ECBs in LPA
- Modifying store-protected Intercomm control blocks in LPA
- Getting, modifying, and freeing SQA and CSA storage
- Making the Intercomm address space non-swapable
- Use of privileged machine instructions for ESS locking
- Use of the MVS high-speed dump facility for system dumps

While these functions are all necessary, it is clear that they must be performed in a controlled environment, where there is no possibility that they can be used to jeopardize system integrity.

3.1.2 NEW SVC FUNCTIONS

The Intercomm SVC routine will be completely rewritten, to implement within it all the required functions currently performed by the existing SVC routine, as well as those functions performed by other Intercomm modules while in key 0 or supervisor state. In addition, the SVC will perform a complete parameter validation whenever it is called, as well as ensuring that all integrity considerations, such as "Time-of-Check-to-Time-of-Use" (TOCTTOU), are adhered to.

All existing SVC "entry points" (as determined by the contents of register 0 at the time of entry to the SVC routine) will be disabled, and their use will result in an immediate termination of the caller with a user abend code of 2048.

The new functions performed by the SVC routine are as follows (a few are functionally equivalent to current functions, but will be implemented in a different, secure, way).

- Region initialization
- Get store-protected storage area
- Get store-protected storage area, and move data into it
- Move data into existing store-protected storage area, owned by caller
- Set protected storage area to all zeros
- Set Intercomm data area in LPA to all zeros
- Free protected storage area
- Move data into Intercomm data area in LPA
- Wait on one or more ECBs in LPA
- Post an ECB in LPA, using cross-memory post
- Set a bit in a protected storage area to 1
- Set a bit in a protected storage area to 0
- Implement the SECLOCK function of ESS
- Implement Intercomm's FASTSNAP function (issue SDUMP SVC)
- Issue SYSEVENT macro, to make caller non-swapable
- Region closedown (free protected areas acquired)

3.1.2.1 PARAMETER VALIDATION

The Intercomm SVC routine will always perform a complete validation of all data areas passed to it, to ensure that only those areas that belong to the caller are being modified, and that only areas that can be read by the caller are being accessed.

It is not possible to validate that the caller of the SVC actually is an Intercomm module[1] (any program could simulate an environment that would fool even the most complicated validation), but it is possible to still adhere to the stated integrity considerations.

NOTE: If Intercomm is running on a system with MVS/SP1 (MVS/370), some of the integrity-related checking can not be done. In such a case, full integrity support will not be available.

DATA AREA VALIDATIONS

Whenever the SVC is called from within a particular address space, it will determine whether this is the first call, or a subsequent call, from this address space. It will make this determination by checking the TCBUSER field in the Job Step TCB. If TCBUSER is 0, then this is the first call, and the SVC environment will be initialized; if non-0, this is a subsequent call and the SVC environment will be verified.

[1] The Program Control Facility of the RACF security package could be used to restrict who can execute Intercomm, and the SVC could use a RACF resource class to validate that the caller is authorized, but the overhead of doing this will be high, the installation complexities will be high, it will not necessarily work with security packages from other vendors, and it will not provide any additional integrity protection.

The SVC environment is initialized by allocating, from a store-protected subpool in the caller's address space, a control block for use by the SVC, initializing this control block, and storing the address of this control block in the TCBUSER field. This control block will be used as an anchor place for all other data areas that the SVC gets on behalf of its caller(s) within the address space.

On every call but the first, the SVC will validate that the TCBUSER field indeed points to a store-protected area and that this area conforms to the structure of the SVC anchor. After this, any other area referenced (e.g., for use as an ECB, or as an Intercomm control block to be modified) must be one of the following:

- A non-protected area (in the caller's address space)
- A store-protected area, chained off the SVC control block (and therefore gotten by the SVC in a previous call)
- Part of an Intercomm LPA module (the MRMCT or SECVECT control blocks), whose addresses were obtained and stored into the SVC control block during the SVC initialization

This validation will ensure that the caller is not trying to use the SVC to reference a storage area that doesn't belong to it, and that it cannot normally access (such as storing into low memory, or accessing a fetch-protected storage area).

In addition, the SVC will not permit any references to a fetch-protected area (Intercomm doesn't use fetch protection for its control blocks, so any such reference must be in error).

SERIALIZATION

When ~~needed~~ To maintain data integrity, the SVC routine ~~will use~~ requires the Local Lock to serialize operations within the Intercomm address space.

In addition, some of the new SVC functions (for example, to wait on several ECBs in protected storage) will require that the caller's parameter data is moved into protected storage while the SVC routine is executing, to prevent its modification by another task within the caller's address space, while the task that actually issued the SVC instruction is waiting for the SVC to complete.

3.2 INVOCATION

The new Intercomm SVC routine will be invoked in the same way as the current routine: by passing parameters in registers, and issuing an Intercomm macro instruction. Although there will be quite a few new parameters defined, the calling conventions do not change.

3.3 ERRORS

The Intercomm SVC routine will validate all parameters passed on all calls to the SVC routine. Any parameter error will result in the immediate termination of the caller, with a user abend code of 2048.

There will be no return codes to indicate individual parameter errors, since no user code is expected to invoke this SVC routine. One exception - if a request to get a protected area cannot be satisfied, the field where the area address is to be stored will contain binary zeros.

3.4 MESSAGES AND CODES

The Intercomm Messages and Codes publication will be updated to reflect the abend code (user abend 2048 or 69) that will be issued by the new SVC routine. A program exception abend code while the SVC routine is executing indicates the SVC was not installed with the local lock, or invalid data area address.

3.5 SECURITY

3.5.1 SYSTEM INTEGRITY

Intercomm will fully support the integrity requirements of MVS. Isogon Corporation will accept, as a valid MSR, any problem report that shows a breach in the integrity of MVS due to the presence of Intercomm, provided that this Integrity Enhancement SM has been applied in unmodified form.

Such MSRs may not be "solved" by issuing a publications change.

3.5.2 EXTERNAL CHARACTERISTICS

The new support will be available only in Intercomm Release 10, when the SVC routine has no installation modifications applied, and when Intercomm is executing on a system under MVS/SP2 (MVS/XA) or MVS/SP3 (MVS/ESA).

4. MIGRATION/COEXISTENCE

The new support will be provided as a set of related, normal system modifications (SMs) to Intercomm.

The new support will only be available under Release 10 of Intercomm. Installations that are still running Release 9 or earlier will have to upgrade to Release 10 in order to benefit from these integrity enhancements. However, earlier releases will continue to be supported in their current form, even after this support becomes available for Release 10.

Note that while these SMs must be applied, like all normal Intercomm maintenance, each installation may elect not to enforce the new integrity validation, by generating Intercomm without the new support and by continuing to use the existing SVC routine. In this case, however, no integrity-related MSRs will be accepted by Isogon Corporation.

This optional support for integrity checking allows those Intercomm installations that have been using the Intercomm SVC for uses other than that for which it was intended (most often to enter supervisor state and/or system key), to continue to do so.

The existing SVC routine will continue to be supported, but only to the extent of correcting non-integrity related MSRs.

5. SUPPORT USE

The proposed change to the Intercomm SVC routine has no external impact, except for the installation of Intercomm.

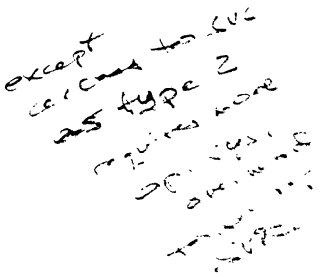
The new SVC routine must be defined as a type 2 SVC, whereas the existing SVC routine is a type 1 SVC.

6. PERFORMANCE

Intercomm, as a high-performance telecommunications monitor, must always operate with a minimum of overhead. This has always been, and continues to be, one of the main criteria when making any changes to the product.

However, when the choice is between providing full integrity and high performance, the integrity considerations must always take precedence. The current SVC routine was designed with performance being of the utmost importance, and it is inevitable that the new routine, because of the additional validation, will have somewhat more overhead. It is, however, expected, that the path length for completely processing a typical transaction (message) that involves use of the MRS and ESS facilities will increase by no more than 1,500 instructions.

In fact, since in many cases the new support will require only one SVC instruction where the current support uses two or more SVC instructions, there may be cases where Intercomm actually will execute faster with the new SVC routine.



7. RESOURCES

7.1 STORAGE

The new SVC routine will require approximately 4K bytes more memory than the current routine.

In addition, the Intercomm modules that call the SVC routine will, overall, grow in size by less than 1K bytes (some routines will actually get smaller, since their code is moved into the SVC routine).

8. RATIONALE

8.1 DESIGN

The design for the new Intercomm SVC routine is based upon the ~~basis~~ premise that the interface to the SVC routine, while internal to Intercomm and not documented in any user publication, can be made public without affecting the integrity of Intercomm or the MVS system as a whole.

This is accomplished by providing enough validation within the SVC routine, by keeping a record of all store-protected data areas owned by (or created on behalf of) the caller, and by ensuring that, once the validation has taken place, there is no "back door" available by which a calling program could affect a change of the parameters, once they have been accepted by the SVC routine.

8.2 ALTERNATIVES

It might seem that one way of preventing unauthorized access to the Intercomm SVC routine, while at the same time requiring less of an implementation effort, would be to use a system security package, such as RACF or ACF2, to control who can execute the Intercomm program.

This, coupled with a requirement that Intercomm is started by a small, APF-authorized, initialization routine, and a check within the SVC routine that the first caller (when the TCBUSER field is 0) must be APF-authorized, would provide an environment that would, from an integrity point of view, be "safe."

However, it would only be "safe" if all the installation-written processing routines were trusted (and error-free). Furthermore, such an implementation would put undue demands on the installations to maintain a list of users that are authorized to start Intercomm, and this would be a problem, especially for those installations that need to stop and start Intercomm at any time of the day or night. It would also make it very hard (and dangerous) to test new Intercomm releases, and new installation-written processing routines.

This approach was therefore abandoned in favor of the present one, where any user can call the SVC routine without any integrity exposure, because:

1. The caller will always receive control back in problem state and the caller's protect key.

2. Store-protected storage may only be acquired, modified, and freed via the SVC routine, which will confirm the caller's ownership.

9. DISTRIBUTION OF PROGRAM PACKAGES

Intercomm has always been distributed in source code form, with ongoing maintenance being applied by the shipment of source code corrections. This has been an acceptable, in fact desirable, form of distribution, especially since many users have applied their own modifications to the product, and need the ability to maintain these changes.

However, such a distribution method is inherently more unsafe than a method that only ships object code (or load modules), since any locally applied modifications may be the inadvertent cause of an integrity problem, especially if Intercomm maintenance is applied without being cross-checked against all local modifications.

The new Intercomm SVC routine will therefore, by default, be distributed in load-module form only, and maintenance will be applied either by shipping a replacement module, or by shipping code changes in SUPERZAP form. Other Intercomm modules, including those that use the SVC routine, will continue to be distributed in source form.

If an installation so desires, it may obtain the source code of the SVC routine by sending a written request, signed by the installation's security officer or auditor. However, Isogon Corporation will not accept any MSRs for problems that are not reproducible with the SVC routine as shipped by Isogon.

This enhancement is available now, and will be shipped to all Intercomm users as part of the normal maintenance procedures for Release 10 of Intercomm.

If you'd like more information about this enhancement, please contact Mr. Per Hellberg, our Vice President of Technology.

Very truly yours,
ISOGON CORPORATION

Gerald Sindler
Vice President