SoftLanding SYSTEMS
A Division of the UNICOM Group of Companies

UNICOM®
UNICOM SYSTEMS, INC.

Supplement #15

84 Elm Street • Peterborough, NH 03458 USA
TEL (010)1-603-924-8818 • FAX (010)1-603-924-6348
Website: http://www.softlanding.com Email: techsupport@softlanding.com

SECURITY RECOMMENDATIONS

TABLE OF CONTENTS

INTRODUCTION

This document provides some security recommendations for your TURNOVER® for iSeries v100 system. You should consider this document to be a supplement to the user guides supplied with your operating system. It's not intended to be a complete description of the security provisions of your system. As a TURNOVER® for iSeries v100 user, you should review the security on all of your production libraries and objects to ensure that changes can only be made through approved channels.

Although you may have authority to some functions of TURNOVER® for iSeries v100 such as to checkout source or to submit a form, you need no special authorities outside of TURNOVER® for iSeries v100. As a programmer or manager, you should not be able to access any more than the object description of production objects.

This document provides some suggestions for setting up security on your system, and describes how TURNOVER® for iSeries v100 uses security to control the promotion process.

The objective of these security recommendations is to ensure that there is a record of every change to production source members and objects in TURNOVER® for iSeries v100.

OBJECT OWNERS

Every object on your system has a designated owner. Our recommendation is that you create a profile that owns all objects within an application and give it a password of *NONE. That way, no one can sign on as that user and assume the owner's object rights. Next, appropriate authority can be given to each user that needs access to the application through authorization lists or on the individual objects themselves.

Another approach is to create an additional User Profile with limited rights to the objects in the application(s) libraries. This second profile can be used as a group profile for all who need to access the application(s). This protects the integrity of both the data and the objects by not allowing excess privileges to any user or developer.

In TURNOVER® for iSeries v100, objects promoted to each application level can have a different owner. When objects are added or replaced, the owner is changed to the profile name you provide (see the ***Application Description*** section in ***Chapter 1: Working with Application Definitions*** of the ***TURNOVER® for iSeries v100 User Guide*** or see the online Help). Object owners can also vary by object type by assigning ownership using reference objects.

We advise against using the profile QSECOFR as the owner of any of your applications. The primary reason for not using QSECOFR is the ability to adopt authority in a program from the object owner. If the program in question presents a command line to the user, they will have all of the authority of QSECOFR.

Furthermore, be sure to limit the authority of your programmers so that they will not be able to update objects in production without the use of TURNOVER® for iSeries v100. Otherwise, you have no control except to the extent that programmers use TURNOVER® for iSeries v100 voluntarily. You will still enjoy the convenience that TURNOVER® for iSeries v100 provides, but the primary benefit of security will be undermined. TURNOVER® for iSeries v100 has the ability to monitor changes not made through TURNOVER® for iSeries v100 by running an audit report. For more about this feature, see ***Chapter 10: TURNOVER® for iSeries v100 Reports*** in the ***TURNOVER® for iSeries v100 User Guide***. TURNOVER® for iSeries v100 also will alert you that a source member or object has been changed outside of TURNOVER® for iSeries v100 when you attempt to check out a source member or object and when you attempt to replace it during a promotion job.

PRODUCTION LIBRARIES

We recommend that you use libraries to separate and secure each application. The ability to define and secure libraries enables you to manage objects for each application more efficiently. You can secure each library to limit access to objects stored there.

UNICOM Systems, Inc. Recommends

After you create your application definitions, run the ***Check Application Definition*** command (**CHKAPPDFN**) to uncover any objects that do not match the object ownership defined for your application. Do this regularly and make sure your technical staff is aware of it.

The **DSPOBJAUT** (**Display Object Authority**) panel might look like this for the library PAYROLL:

	Object	**----Object--------**			**-----------Data-----------------**			
User	**Authority**	**Opr**	**Mgt**	**Exist**	**Read**	**Add**	**Update**	**Delete**
QSECOFR	*ALL	X	X	X	X	X	X	X
PAYMGR	*USR DEF	X	X	_	X	X	X	X
PAYDEPT	*USE	X	X	_	X	_	_	_
*PUBLIC	*EXCLUDE	_	_	_	_	_	_	_

The Owner is PAYMGR. He has altered his own authority to the library so he does not inadvertently delete the library (Exist). He can maintain authority to the library for others, including QSECOFR, if he desires. He can change his own authority. He has all rights to all objects within the library.

In this example, PAYDEPT is a group user. All members of the payroll department share the authority of user PAYDEPT (each as GRPPRF(PAYDEPT) in his/her user profile). (Alternatively, an authorization list or a list of individual user profiles could be used.) Everyone within each group has access to the library, but members of the group can still be restricted from specific objects within the library.

If a particular program within the application is designed to add or delete an object (such as a file member), that program could be changed to adopt the authority of the object owner (in this case PAYMGR) to enable the add or delete operation to take place.

On the iSeries, *PUBLIC is specifically excluded to ensure that, no matter how object authority was set up, only users authorized at the library level could access objects in the library. The user would see a "Not authorized to library" message, and could not get into any object in the library no matter how the object authorities were defined.

When viewing the **DSPOBJAUT** panel for a library, keep in mind that “Data” means entities within the object displayed; in this case, the library. In other words, “Data” rights are rights to the objects within the library, not the data within those objects. You need to grant authority to objects within the library to grant data rights. (See recommendations in the following section).

SETTING SPECIFIC OBJECT AUTHORITY

Generally, objects are of two types: production files (physical and logical files) and program-related objects (programs, message files, display files, and so on).

To set up authority reference objects for use during form processing, see ***Setting up authority reference objects*** on page 7.

Production files

Specific authority granted to a user, or groups of users, is usually ***CHANGE** (iSeries) and includes:

	Object	**----Object--------**			**-----------Data------------------**			
User	**Authority**	**Opr**	**Mgt**	**Exist**	**Read**	**Add**	**Update**	**Delete**
OBJUSER	*CHANGE	X	_	_	X	X	X	X
OBJOWNER	*ALL	X	X	X	X	X	X	X

The object owner has all rights to the object. The object users usually have:

- Operational rights
- All data rights (Read, Add, Update, Delete)

Programs and related objects

Specific authority granted to a user, or groups of users, is usually ***USE** (iSeries) and includes:

	Object	**----Object--------**			**-----------Data------------------**			
User	**Authority**	**Opr**	**Mgt**	**Exist**	**Read**	**Add**	**Update**	**Delete**
OBJUSER	*USE	X	_	_	X	_	_	_
OBJOWNER	*ALL	X	X	X	X	X	X	X

The object owner has all rights to the object. The object users need only:

- Operational rights
- Read rights

UNICOM Systems, Inc. Recommends

Secure your objects with authority lists instead of using special authority; these are easier to maintain. We recommend that you not mix authority lists and special authorities—performance will suffer because of the additional checking done by the operating system.

Source files and members

Production source files need to be secured as well. There is both library and file authority to consider.

If you keep production source files in a separate library, we recommend that you grant authority as follows:

Source library

	Object	**----Object--------**			**-----------Data------------------**			
User	**Authority**	**Opr**	**Mgt**	**Exist**	**Read**	**Add**	**Update**	**Delete**
QSECOFR	*ALL	X	X	X	X	X	X	X
QPGMR	*USE	X	_	_	X	_	_	_

The programmer cannot add any files or members to the library. S/he would need ‘Add’ rights for the library to be able to add a member. The source files in the production library should be secured as follows:

Source Files

	Object	**----Object--------**			**-----------Data------------------**			
User	**Authority**	**Opr**	**Mgt**	**Exist**	**Read**	**Add**	**Update**	**Delete**
QSECOFR	*ALL	X	X	X	X	X	X	X
QPGMR	*USE	X	_	_	X	_	_	_

Programmers cannot add or update members. They can only view the member list and view the contents of a member. If your programmers are not part of group profile QPGMR, then secure them by authority list or by specifying their user profiles in the object authority list.

Unfortunately, you cannot set file authority to allow programmers to see a list of files and members, but not their contents. If you do not want programmers to be able to view a member list or the contents of a source file other than through TURNOVER® for iSeries v100, then you could remove the ‘Read’ rights to the source files. This would still be adequate for checking out source by object, where you would work with an object list in PDM and check out the source for an object using a TURNOVER® for iSeries v100 user-defined command.

Setting up authority reference objects

We suggest that you set up dummy objects (REF*xxxxxx* where *xxxxxx* is PF, LF, RPG, CLP, MSGF, DSPF) in one or more production libraries. Then, when you define the ***Create parameters*** and ***Authority*** parameters in the Line Defaults of the application definition, you can reference these objects by setting the ***Reference object*** parameters to **REF***. TURNOVER® for iSeries v100 references the named object when it grants authority to objects you submit to be promoted. They can reside in any library you choose.

If you use **REF*** in the ***Reference object*** portion of the ***Create parameters*** and ***Authority*** fields, TURNOVER® for iSeries v100 parses together “REF” with the object type attribute code, such as “PF” or “CLP” and then looks for a reference object with that name, for example “REFCLP”. Once you take the time to create reference objects,[1] TURNOVER® for iSeries v100 maintains object authorities for you. We’ve created several objects in library SOFTTURN that you can duplicate to get started:

REFPF	a reference physical file
REFLF	a reference logical file
REFRPG	a reference program
REFCLP	a reference program
REFCMD	a reference command
REFMSGF	a message file
REFDSPF	a reference display file
REFPRTF	a reference print file
REFDTAARA	a reference data area

Although TURNOVER® for iSeries v100 permits you to reference any object on your system, if you use object types other than those on the list, be sure to create your own “REFxxxxxx” objects so that you can use the **REF*** feature consistently in all of your applications.

Object Distribution and Security

When TURNOVER® for iSeries v100 distributes objects, it does not transfer private authorities to the destination computer. The OS/400 operating system retains private authorities with user profiles, not with objects. Therefore, in TURNOVER® for iSeries v100, you cannot use a ***Create parameters*** setting of **T** (Test) on a remote computer or preserve private authorities by moving an object to a remote computer. Additional information about how OS/400 handles private authorities and objects follows:

- If user profiles have private authority to an object that’s being restored, those private authorities are usually not affected. (However, restoring certain types of programs can cause private authorities to be revoked.)

[1] For a shortcut to creating reference objects and other missing objects for an application, see ***Chapter 1: Working with Application Definitions*** (in the ***TURNOVER® for iSeries v100 User Guide***) and the explanation of **F6=Create missing items** on the application definition’s ***Type Codes*** panel.

- If an object is deleted from the system and then restored from a saved version, private authority for the object no longer exists on the system. When an object is deleted, all private authority to the object is removed from user profiles.

- If you need to recover private authorities, you must use the **Restore Authority** (**RSTAUT**) command. The normal sequence is:

  Restore user profiles
  Restore objects
  Restore authority.

CHECKING YOUR SYSTEM

Here are some suggestions for checking your system for security exposure.

Check the application definition reports

After you have defined your application(s) to TURNOVER® for iSeries v100 you can run this report (see TURNOVER® for iSeries v100 Reporting Menu) to show you ownership, authority and other potential problems you may have to correct. You can either run it before using TURNOVER® for iSeries v100, or at any later time. Once you're sure that the application definitions are correct, you should run the ***Check application definition*** (**CHKAPPDFN**) command to highlight any problems.

Physical security

Controlling physical access to your computer and workstations is the first defense against unauthorized access to your system.

Password security

Require password changes frequently, and prohibit users from telling others their passwords. Use the password change procedures that are part of the operating system.

Device security

Users can be authorized or excluded from specific workstations. You should be sure to limit QSECOFR to only the most essential workstations.

Menu security

We recommend that you employ a menu management system to control which functions your users can perform when they sign on. Being able to easily change users' options eliminates the temptation to "borrow" someone else's password.

Security level system value

If you haven't done so, change the system security level (System Value QSECURITY) to "30" or higher. Any value lower than "30" grants all users all rights to all objects! If you upgrade the security level to 30 or higher, be sure to read the instructions in your iSeries Security Guide before doing so.

Review special authorities

You should set special authorities (those specified when you create a user profile) appropriately for each user. We suggest that you review all users who have special authority other than ***USER** and determine if the authority is appropriate for that user. You can display user profiles to a file and query that file.

Review group profiles

We suggest that you review all user profiles that have a group profile specified to determine if the group specified is correct. Carefully review any user who is part of group QSECOFR, QPGMR, or QSYSOPR. These constitute a potentially serious security exposure. You should also review the authority for the group profiles to ensure that they are appropriate.

Check adopted owner authority

Review all programs that adopt owner authority—especially those that adopt QSECOFR or other system user profiles.

Use the **DSPPGMADP** (Display Program Adopt) command to display programs that adopt owners, specifying user profile of owners of sensitive objects on your system. (Start with QSECOFR.)

Check user profile menu

Check all user profiles that have a menu specified. We recommend that you use ***SIGNOFF** for the menu parameter, except where necessary. If a user presses **F3** from a main menu that is an initial program, the system will display the menu specified in the user profile.

Output authority

Authority to report output should also be considered. Test authority to output queues and spool files to ensure that only authorized persons can view report or text output on your system.

Questions to ask yourself about security on your system

This is just a sampling of some of the questions you might be asked during a security audit. We've included some explanatory text in italics to help you consider the implications of your security policies.

1. Who has access to user profile QSECOFR, or to programs owned by QSECOFR that adopt owner's authority?

   If the number of people who have access to the QSECOFR profile is limited, then so is the likelihood that unauthorized access to programs and objects will occur. Only the owner of an object and people who use it should be granted authority to that object.

2. Who has *ALLOBJ authority?

   *Regular TURNOVER® for iSeries v100 users should not have *ALLOBJ authority. Even user profile TURNOVER does NOT need *ALLOBJ authority for TURNOVER® for iSeries v100 to work. In fact, no user other than QSECOFR needs *ALLOBJ authority for TURNOVER® for iSeries v100 to work, because strategic TURNOVER® for iSeries v100 programs adopt QSECOFR authority.*

3. Who has explicit authority to production data and program libraries?

   Just as above, regular TURNOVER® for iSeries v100 users should not have explicit authority to production data and program libraries, and user profile TURNOVER does NOT need explicit authority for TURNOVER® for iSeries v100 to work.

   *Many TURNOVER® for iSeries v100 programs that are USRPRF(*OWNER) are owned by TURNOVER, rather than QSECOFR. This is necessary so that TURNOVER® for iSeries v100 can update the TURNOVER® for iSeries v100 database as necessary without giving TURNOVER® for iSeries v100 users explicit authority to the files (which would enable them to update the files by means other than TURNOVER® for iSeries v100).*

   If your auditors have a problem with user profile TURNOVER owning objects, then you can change user profile TURNOVER so as not to have a password (this will not cause any problems). However, the TURNOVER profile must have authority to update the TURNOVER® for iSeries v100 files.

4. What programs on your system currently adopt owner's authority?

   You should audit any programs on your system that adopt owner's authority.

5. Do programs that adopt authority call other programs? If so, is the ***Use Adopted Authority*** parameter for the called program set to **No**?

   *USEADPAUT(*YES) does NOT cause a program to adopt authority. It just means that if the program is called by another program which is USRPRF(*OWNER), the adopted authority of that program is not dropped. In other words, if a program is USEADPAUT(*YES) and it's owned by QSECOFR, and you call that program from a command line, you will NOT be adopting QSECOFR authority. USEADPAUT(*YES) is the IBM-shipped default for the compile commands.*

   The TURNOVER® for iSeries v100 programs that adopt QSECOFR authority all have their own internal authority checks to make sure the user running the program has the appropriate authorities defined within TURNOVER® for iSeries v100.

6. Do programs that adopt authority present a command line to the user? If so, what is the authority of the program owner?

   Anywhere that a user can enter a command within TURNOVER® for iSeries v100 (i.e. the Programmer Worklist command line, an ***F21*** *command line, or the command line in SEU) adopted authority is temporarily dropped (a program is in the stack that is USEADPAUT(*NO)).*

7. Who has access to the **CHGPGM** command, **CHGOBJOWN** command, and the **CHGSYSLIBL** command?

   CHGPGM: *The* ***User profile*** *and* ***Use Adopted Authority*** *parameters of the* ***CHGPGM*** *command can be used to circumvent established security requirements.*

   CHGOBJOWN: *This command can allow people to change object ownership, which can permit unauthorized access to objects.*

   CHGSYSLIBL: *If your programmers or users can modify their job's system library list, they could place a program higher in the library list than a product library. This creates a security exposure. IBM ships this command to you with authority limited to* ***QSECOFR***.

There is no security magic. If it's really important at your company, security is not something you can just set and forget. The solution is more likely to be found in continually monitoring your security exposure and understanding where the potential leaks may be.

If you have any questions about the information in this document, please contact a UNICOM Systems, Inc. Technical Support Representative by phone, fax, or email at the locations shown at the beginning of this document.