# 4065

Highly granular OLTP application security

Steve Schiefelbein
UniKix Technologies
8125 N. 23rd Avenue, Suite 195
Phoenix, Arizona USA 85021
602-242-3300

UniKix Technologies is a supplier of middleware to run large enterprise OLTP applications on UNIX. Our core product, UniKix, is a CICS-compatible OLTP system on UNIX, and so our software is often used by banks, insurance companies, retailers or manufacturing companies who want to run their mission-critical OLTP applications on UNIX. The financial nature of many of these applications has created a strong interest in additional security facilities from many of our customers. This paper will show how we took advantage of CA-Unicenter, an external security management product, to provide more robust, secure applications for our customers.

Market Evolution

UNIX was initially created as a multi-user time sharing system, and many early users followed that profile. In the 80’s, companies started to use UNIX for more commercial applications. The rapid growth of relational database usage on commodity platforms pulled UNIX heavily into the commercial world, as well as growth in the use of UNIX as a workgroup server or for departmental applications. All of these uses created the demand for systems management tools for UNIX, including better user-level security support. Systems management tools such as CA-Unicenter focused their security functions for these types of users as well, providing extended security controls for UNIX file access for relational databases and for setting UNIX privileged user Ids.

While these functions were excellent for traditional usage, they didn’t provide the granularity of control needed for large multi-user transaction processing applications. Transaction processing systems, typically, involve hundreds or thousands of users accessing hundreds of transaction programs, all accessing or updating the same databases. This creates the need to control who can use which transaction type, whether they can access the database and/or update, which database can be accessed by which user - and so on. Traditionally, large OLTP applications ran as CICS transactions on IBM mainframes, where products such as IBM’s RACF and CA’s Top Secret provided security controls on individual assets or resources within the OLTP application, and provided this level of granularity. In the 90’s, many economic and technology factors combined to cause a shift away from mainframes as the best choice for large OLTP applications, and toward open client/server systems based on UNIX servers. These factors included:

Productivity - almost all new productivity tools appear first on open systems, not the mainframe

Cost - mainframe systems cost roughly 10 times that of a UNIX server of the same power

UNIX robustness - UNIX itself matured, and with products like CA-Unicenter became roughly equal to the mainframe in robustness and manageability.

The terminal network deployed for mainframe configurations was typically a closed SNA Token-Ring network using 3270 terminals, which minimized the security concerns for the mainframe OLTP applications. That network typically would have terminals in fixed, known locations, probably somewhere on the company property, with a reasonably well-known group of operators using them to access those mainframe OLTP applications. However, this type of network is no longer satisfactory for the increasing connectivity needs companies now face. The types of users needing access to these OLTP applications from outside a private network continues to grow. Open TCP/IP networks provide the most effective means for this needed connectivity. However, such networks are susceptible to security compromise due to the nature of the TCP/IP network protocol, and network firewalls are needed to guard against these outside threats. There are a variety of firewall products for TCP/IP networks that provide security from outside malicious attack such as password sniffing, IP spoofing, session hijacking, protocol bugs, and denial-of-service attacks. These products ensure the authenticity of users that access the system and its applications: "you are really who you claim to be". These authentication challenges of the TCP/IP protocol arise from its use of "well-known" ports where applications such as telnet, sockets, ftp, e-mail, and now UniKix "listen" for requests.

Companies now implement Web Intranets to serve internal company communication needs, and may also desire a presence outside the company on the Internet. They may also need to interact with other companies via TCP/IP networks. This increasingly exposes company-sensitive information and mission-critical OLTP applications to unauthorized access from within as well as from outside the company, and requires significant investment by the company in developing an enforceable security policy with the hardware and software to support that policy.

UniKix is the CICS-on-UNIX product that supports offloading of mainframe applications, and thus supports the connectivity possibilities that TCP/IP provides. Access from PC's, UNIX workstations, X-terminals, and even from the Internet (using WebKix) is supported along with existing 3270 terminal networks through SNA gateways. Of course, such a wide variety of connectable sources further magnifies the need for a sound, enforceable policy and software support in UniKix and UNIX. These connections need to be included in the security policy implementation.

More robust, policy-based security support is needed. Products such as RACF and CA-Top Secret provide such support on the mainframe. These products provide for defining the rules for the company security polices, and provide the enforcement within CICS by denying access to unauthorized users. Several UNIX software products such as CA-Unicenter provide the definition and administration tools necessary to implement the software support for security policies, and several "firewall" products provide the necessary hardware/software for user authentication support.

CA-Unicenter enhances standard UNIX security by providing policy-based rule definition and enforcement, extending the standard UNIX file permissions and providing for definition of UniKix- and CA-specific assets and permissions to those assets. Users can be configured with a variety of password controls, including timed passwords, required minimum / maximum password change intervals, and calendar-based accessibility. The UNIX superuser is the most prone to abuse / attack, and CA-Unicenter has provided the ability to limit the permissions that superusers can have; this then provides protection for multiple administrators on the same UNIX system as well as limiting the ability for malicious attacks to do significant damage. CA-Unicenter provides configurable logging of access violations, as well as userid lockout rules due to violations.

Note, however, a sizable proportion of security compromises occur from within the company (bypassing security in order to get something done, applications not sufficiently tested, exposing passwords), because users are not educated about the company security policies and the importance of adhering to them. Thus, any security implementation requires education and "buy-in" to the importance of adhering to the policy rules that are developed.

Securing OLTP applications

As customers started to migrate these OLTP applications to UNIX, they began to call for the same level of granular control of resources for UNIX-based OLTP applications as was possible on the mainframe. UniKix Technologies, as a UNIX OLTP application developer, responded to this call and defined the new assets or resources users needed to control for security purposes. These assets were then added to CA-Unicenter. With the additions, UniKix Technologies could use the functions in its UniKix OLTP system to check these assets during normal operation.

UniKix and CA-Unicenter jointly defined highly granular asset definitions and controls to be added to CA-Unicenter after agreeing on which resource or asset controls would most benefit the large corporate OLTP customers. These assets include:

- Specific UniKix transactions that a user can access.
- Specific UniKix programs within a transaction that a user can access.
- Specific VSAM indexed sequential files that can be accessed and whether this is a read or write access.
- Access to Temporary Storage and Transient Data Queues for CICS transaction to transaction communication.
- Whether data about specific transactions can be journalized for accounting and billing purposes.
- Controls that tie individual users to selected physical terminal devices and that control which transactions can be run from these devices.

Implementation was a parallel effort with development occurring in both the CA-Unicenter product and in the UniKix product. The additional asset types were added within CA-Unicenter. The UniKix OLTP system links to CA-Unicenter when transactions request access to these CA-Unicenter controlled assets.

CA-Unicenter supports UniKix-specific assets that are equivalent to RACF / CA-Top Secret resources, and thus provides the foundation for UniKix to support the equivalent mainframe class security. These controls would give the granularity of security control that major customers required, especially those in financial services.

Security needs within UniKix

UniKix security needs are similar to mainframe CICS, and include:

Unsecured terminal locations, where only the users authorized to submit transactions from those locations can be sufficiently protected from unauthorized users employing those terminals.

Telnet users, which are users that connect to the system from unknown locations and thus must be authenticated as valid users to UniKix.

Web users, who can submit requests via WebKix to the UniKix as “anonymous” users; these can be Web users either from an Intranet or the Internet, depending on how the Internet firewalls have been established.

Multiple administrators; UniKix allows for administrative functions to be done by authorized users, but needs to be able to subset those functions to the appropriate administrator. E.g. all administrators can inquire on the state of terminals, but only the terminal administrator should be able to enable / disable any of the terminals.

Rogue applications, which are insufficiently tested applications that compromise the company’s UniKix system. UniKix needs to be able to prevent users from accessing any resources that they do not need access to, and thus prevent such a rogue application from causing harm.

Without CA-Unicenter, UniKix does provide CICS-equivalent user login support, via CSSN / CESN system transactions and the Sign-On Table (SNT), with passwords if so configured. UniKix assumes the user session as authenticated if the user is accessing UniKix from a UNIX logged-in session. But without CA-Unicenter, UniKix does not provide support for ensuring that a specific terminal can only be accessed / signed on to a specific user / set of users, a.k.a. point-of-entry validation. Any such checking is the responsibility of application coding, for example, with a customer-written login transaction launched when a user connects to UniKix.

UniKix employs (and provides, via CESN/CSSN) user authentication and policy-based access authorization services of CA-Unicenter to address these needs and provide true policy-based transaction and resource security rule enforcement. Rules like "Marketing users are only allowed to query Sales History data files" can be defined through CA-Unicenter user groups and UniKix asset groups. For example, users A, B, and C comprise the Marketing group, and they are defined to have only Read permission to the VSAM file SalesHistA, even though they can be permitted to run transactions X, Y, and Z which other users can use to update that VSAM file. UniKix also will require CSSN / CESN passwords as configured on the CA-Unicenter security database. And, point-of-entry enforcement can be easily done with CA-Unicenter rules by defining a UniKix terminal as accessible only by a specified user / set of users.

Secured UniKix resources

UniKix uses CA-Unicenter security to provide endpoint security; the groups of users who can access a specific UniKix application region name and from what UniKix terminals.

UniKix uses CA-Unicenter to verify that users have permissions to connect to UniKix. A user must have permission to access the UniKix region name, which is configured through the CA-Unicenter UNIX-APPLS asset; this controls who can start that UniKix region name, who can terminate that UniKix region name, and who can connect to and run transactions on that UniKix region. For example, the 'kixadmin' user may be configured to be allowed to start and terminate UniKix region 'production', where the Marketing user group is configured only to connect to and run transactions on 'production'.

UniKix uses CA-Unicenter also to verify that users have permissions to connect to that UniKix region name from the UniKix terminal they are connected with. Users can be configured with permission to a specific UniKix terminal name through the CA-Unicenter KIX-TERMINALS asset. For example, the Marketing user group may be configured with permissions to only connect to UniKix region 'production' from UniKix terminals MK01, MK02, or MK03, and to be denied access from any other UniKix terminal. This also then provides point-of-entry security.

UniKix uses CA-Unicenter security to provide policy-based access control on transactions submitted by users of the UniKix region. Users are configured permissions to execute a given transaction-id through the CA-Unicenter KIX-ATTACH-TRANS asset. For example, the Marketing user group can be configured with permission to execute the SALE transaction (where they inquire on sales history), but configured to be denied permission to execute any other transactions, like CEMT, CSMT, etc. which are UniKix system administrator transactions.

UniKix uses CA-Unicenter security to provide resource-level access control for CICS-equivalent resources: VSAM files, application programs, accounting and user journals, transient data queues, temporary storage queues, terminal destinations, and spawning new transactions. UniKix requests verification of permissions for each resource accessed by a transaction running on behalf of a user. If denied, UniKix will (typically) return a NOTAUTH status to the EXEC CICS statement in the application, which allows for application logic to deal with the access denial.
CA-Unicenter provides for configuring the type of access permitted for a user on a UniKix;

for a specific VSAM file, read access only, or write access also;
for a specific application program, LOAD or XCTL / LINK allowed;

for spawning a specific new transaction-id, whether CANCEL is permitted or only START / DELAY;
for a specific temporary storage queue, read access only, or write access also.

For the other UniKix resources, a single permission type is provided. E.g. any access to a specific transient data queue involves an update of the queue, so only write access permission is checked.
For example, the Marketing user group can be configured for read access only to the VSAM file SalesHistA; this permission is checked in all transactions that access that SalesHistory file, and thus no application can erroneously update it when its transaction is executed by one of the Marketing users.

These resource access controls provide a detailed level of security protection from unauthorized users, beyond that provided with just transaction access controls through the KIX-ATTACH-TRANS asset. This type of security protection prevents users from running transactions that access resources for which those users are not authorized, based on the implemented security policy.

Without this type of access control, it would be difficult to show that a rule such as “Marketing users are only allowed to query SalesHistory data files” is being enforced throughout the UniKix applications. This also relieves the application code from the responsibility for such security checking and puts it in the hands of the CA-Unicenter security administrator.

Resource access controls also provide protection from “rogue” applications accessing the incorrect resource improperly, which could happen when an application is not properly tested before being released into production. For example, suppose the SALE transaction was changed and a bug introduced that would generate an update to the SalesHistory file on all requests, including inquiries! A Marketing user running a normally valid inquiry would see the SALE transaction fail with a NOTAUTH status reported, instead of having it clobber the SalesHistory file.

Finally, UniKix with CA-Unicenter security provides access controls for its CEMT administrative transaction. UniKix (and its CICS counterpart on the mainframe) allows a user with permission to run a CEMT transaction to be able to do any and all operations provided by that transaction. This is much like the UNIX superuser permission problem, where more power is given than the security policy would prescribe. CA-Unicenter provides for defining permissions to specific functions within CEMT through the KIX-COMMANDS asset. Each type of CEMT ‘resource’ is identified with a name used by UniKix to check access permissions:

TASK	for any TASK, FACILITY, etc. function
PROGRAM	for any PROGRAM function
TERMINAL	for any TERM, TERMINAL, etc. function
CONNECTION	for any CONNECTION function
TDQUEUE	for any TDQUEUE or DEST function
FILE	for any FILE function
SYSTEM	for any SYSTEM function

To INQUIRE on any of these CEMT resources then requires read permission for the requesting user, and to SET requires write permission.

This then allows the CA-Unicenter security administrator to restrict which users running CEMT can inquire / change terminal status, task status, etc.

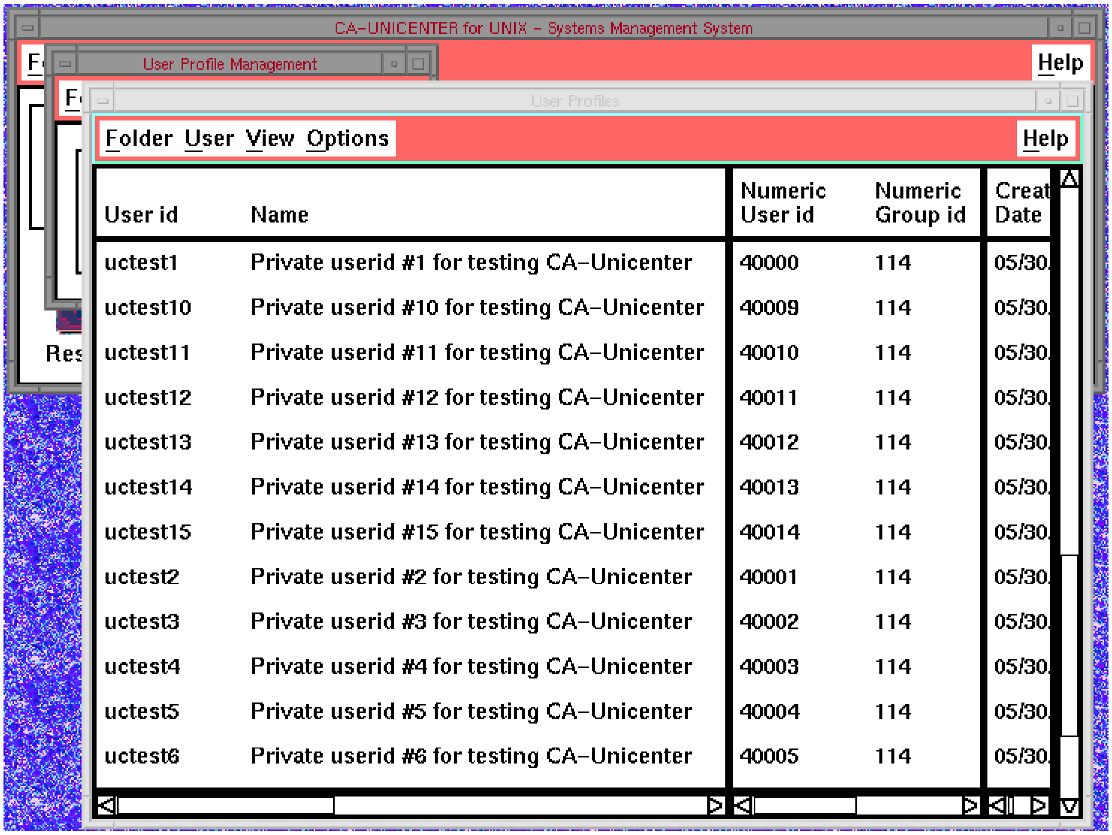
Customer Benefits

The new application assets are available for definition via the standard CA-Unicenter graphical user interface. Similarly, the UniKix system administration can elect these options for the transaction applications they are managing. The resulting combination of CA-Unicenter and UniKix now exceeds the level of security provided by CICS on the mainframe and closely parallels the functions provided by RACF or Top Secret.

With this new level of granularity customers can control who can use which transactions, whether these transactions can read or write critical files, and even where the transaction can be executed from. Now, a banking customer planning to offer a limited number of transactions to the public from a specific banking kiosk in a trial area would have all the tools needed to set up these controls.

Conclusion

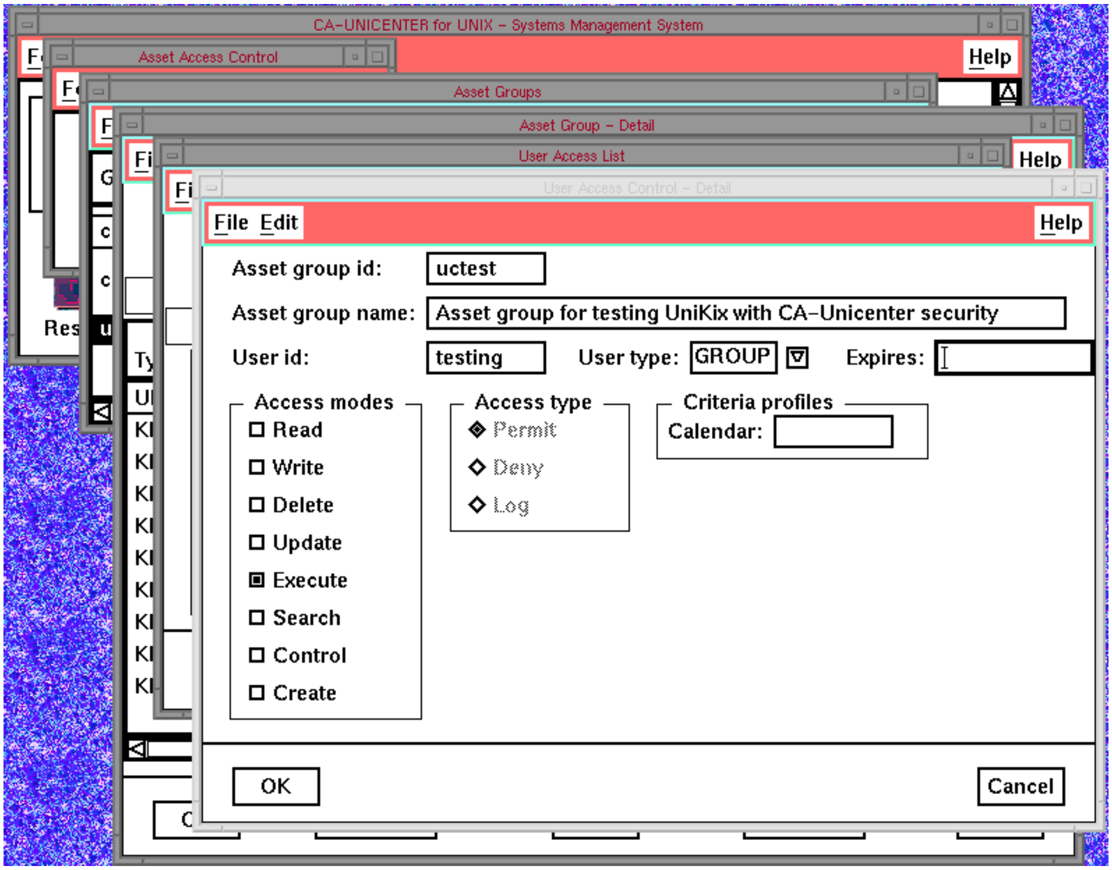
CA-Unicenter with UniKix provides a major step forward in security control for large OLTP applications on UNIX. Customers can get OLTP application level security at the same level of granularity as on the mainframe and can control all the major assets the application uses from transactions to files to communication regions.



User id	Name	Numeric User id	Numeric Group id	Creat Date
uctest1	Private userid #1 for testing CA-Unicenter	40000	114	05/30
uctest10	Private userid #10 for testing CA-Unicenter	40009	114	05/30
uctest11	Private userid #11 for testing CA-Unicenter	40010	114	05/30
uctest12	Private userid #12 for testing CA-Unicenter	40011	114	05/30
uctest13	Private userid #13 for testing CA-Unicenter	40012	114	05/30
uctest14	Private userid #14 for testing CA-Unicenter	40013	114	05/30
uctest15	Private userid #15 for testing CA-Unicenter	40014	114	05/30
uctest2	Private userid #2 for testing CA-Unicenter	40001	114	05/30
uctest3	Private userid #3 for testing CA-Unicenter	40002	114	05/30
uctest4	Private userid #4 for testing CA-Unicenter	40003	114	05/30
uctest5	Private userid #5 for testing CA-Unicenter	40004	114	05/30
uctest6	Private userid #6 for testing CA-Unicenter	40005	114	05/30

Figure 1:

CA-Unicenter “User Profiles” screen showing example of UniKix user-id’s



Asset group id: uctest

Asset group name: Asset group for testing UniKix with CA–Unicenter security

User id: testing User type: GROUP Expires:

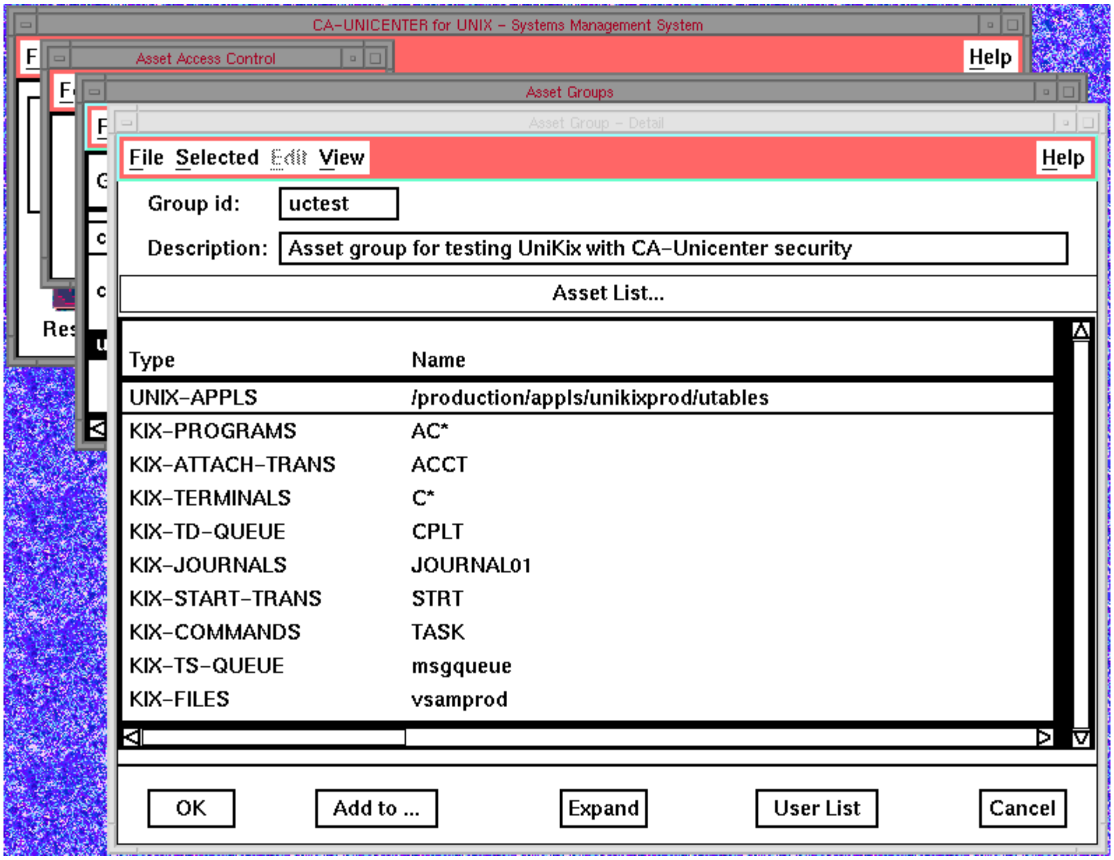
Access modes
- □ Read
- □ Write
- □ Delete
- □ Update
- ☒ Execute
- □ Search
- □ Control
- □ Create

Access type
- ● Permit
- ○ Deny
- ○ Log

Criteria profiles
Calendar:

Figure 2:

CA-Unicenter “User Access Control” screen for UniKix “testing” group



Group id: uctest

Description: Asset group for testing UniKix with CA–Unicenter security

Type	Name
UNIX–APPLS	/production/appls/unikixprod/utables
KIX–PROGRAMS	AC*
KIX–ATTACH–TRANS	ACCT
KIX–TERMINALS	C*
KIX–TD–QUEUE	CPLT
KIX–JOURNALS	JOURNAL01
KIX–START–TRANS	STRT
KIX–COMMANDS	TASK
KIX–TS–QUEUE	msgqueue
KIX–FILES	vsamprod

<u>Figure 3:</u>

CA-Unicenter “Asset Group” screen showing example UniKix assets