insure/SECURITY

User Guide and Reference

Version 8.0

Document date: 09/15/2011

Centerfield Technology, Inc.

http://www.centerfieldtechnology.com

Notice: The information contained in this document is subject to change without notice. Centerfield Technology, Inc. shall not be liable for errors contained herein or consequential damages in connection with the furnishing, performance, or use of this material.

The sample source and resulting programs are provided for your use only. You may repackage any source or resulting program or portion of source or resulting program provided within this document for the purpose of product distribution, installation, or system archiving. Any other use requires the written permission of Centerfield Technology, Inc.

Use of this book and the information within implies an understanding and agreement with the above statements.

Centerfield Technology, Inc.
3131 Superior Drive NW - Suite C
Rochester, MN 55901

1 Overview of HomeRun

The HomeRun toolset allows you to efficiently design, deploy, manage, and support all aspects of an SQL-based environment. The set of tools in HomeRun includes:

insure/INDEX
insure/ANALYSIS
insure/MONITOR
insure/RESOURCES
insure/SECURITY

All tools make use of the HomeRun server on the iSeries. The password you enter on the server controls which tools are available for your use. However, the manual for each of these tools is included in your product download.

In addition, the following tools are included with every HomeRun installation:

Autonomic DBA
Visual SQL Explain (a.k.a. sql/OPTIMIZER)
Database Explorer
Lock Detector
Semaphore Wait Detector
Mutex Wait Detector

See the Appendixes for more information on using these built-in tools.

1.1 Product Contents

The product you received from Centerfield Technology, Inc. contains the following items:

1.1.1 HomeRun installation

The download contains the software for both your iSeries and your Windows 2000/XP/Vista/7 client workstation. Refer to the Product Installation section for instructions on installing the software.

Also, for JDE OneWorld® (aka Oracle E1®) environments there is an additional piece of software that needs to be installed on the Windows® Terminal Server (i.e. Citrix® server).

1.1.2 User Guide and Reference

Manuals for all tools in the HomeRun toolset can be found in your product download. Each manual is intended to help you get started with the tool, explain the tool's features, and provide guidance on the effective use of the product.

1.2 Product libraries

Centerfield uses a naming convention where all IBM System i™ object names are prefixed with the letters "XC".

As part of that naming convention throughout this manual the Centerfield ***{program library}*** will be "XCENTER80" and the Centerfield ***{data library}*** will be "XCENTERD80".

This version of the product uses TCP/IP port number 9920 by default. The rest of this document refers to this port as the default port "**{default port}**". If that port is in use, the installation will incrementally search for unused port numbers starting at **{default port}** and continue looking until one is found. To determine the port that was chosen use the following command:

```
WRKSRVTBLE SERVICE(CENTERFIELD_SERVER_80)
```

2 Introduction to insure/SECURITY

insure/SECURITY is intended to let you secure data when users access your system through ODBC, JDBC, and various Client Access interfaces.

NOTE: To use insure/SECURITY support for remote interfaces (such as ODBC, DDM, or FTP), you must enable remote monitoring with the use of the ADDMON command in the *{PROGRAM LIBRARY}* library. See the section on ***Enabling Remote Monitoring*** for further information.

3 Getting started with insure/SECURITY

This section describes how to get started quickly with insure/SECURITY. It contains basic usage information for users who want to get started right away. For in-depth information about advanced features of insure/RESOURCES, see the next section.

The steps outlined in this Getting Started section are:

1) Install the product
2) Use the ADDMON command to enable remote monitoring
3) Review the security issues associated with remote interfaces
4) Set up your security scheme using the insure/SECURITY tool. This step involves setup for one or more of the following:
   a) Controlling use of a remote interface globally
   b) Restricting use of a remote interface by time of day
   c) Managing use of a remote interface by individual user
   d) Creating proxy users for remote interfaces

3.1 HomeRun requirements and installation

3.1.1 Product Requirements

The HomeRun toolset requires specific System i and Windows hardware and software before it will install and perform correctly.

Here is the list of required IBM PTFs:

- ✓ V5R4:
  - MF39418, MF39419, MF39466, SI24100, SI23714, SI23713, SI24580, SI23891, SI23890, SI24569, SI24504, SI23485, SI25668, SI25041, SI23365, SI23514, SI28951, SI28952, SI28953, SI30076, SI30013, SI30014, SI31631, SI31633
- ✓ V5R4M5:
  - SI24100, SI23714, SI23713, SI24580, SI23891, SI23890, SI24569, SI24504, SI23485, SI25668, SI25041, SI23365, SI23514, SI28951, SI28952, SI28953, SI30076, SI30013, SI30014, SI31631, SI31633
- ✓ V6R1:
  - SI31727, SI31641, SI31407

NOTE: PTFs listed above are current with the release date of this document. Centerfield may learn of new PTFs after the document release so we strongly encourage you to check our website for latest PTF requirements – www.centerfieldtechnology.com

3.1.1.1 System i

- ✓ i5/OS V5R4, or V6R1, V7R1
- ✓ TCP/IP configured

3.1.1.2 Windows

- ✓ Windows 2000, Windows XP, Windows Vista, Windows 7
- ✓ Minimum 80486 @ 66Mhz, 32 MB RAM
- ✓ 50 MB of disk space for product
- ✓ TCP/IP installed and configured
- ✓ ODBC driver installed

3.1.1.3 Citrix Windows Terminal Server for E1

- ✓ Windows NT Server, Windows 2000 Server, Windows 2003 Server

3.1.2 Product Installation

The HomeRun toolset has software that needs to be installed on the System i and on the workstations. The installation processes for both the System i and the workstations are designed to be simple interfaces that provide good default values. The following sections in combination with the on-screen documentation help you understand the installation processes.

If you're re-installing HomeRun you **MUST** refer to the **"HomeRun Installation Guide"** document rather than these 1st time installation instructions.

3.1.2.1 System i Installation

1) Before beginning the installation on the System i, please review this list of the impacts the installation will have on your system. For more detailed information about the HomeRun System i installation, see the Appendix titled *System-Wide Installation Impacts*.

✓ This product uses TCP/IP to communicate between the System i and the personal computer. TCP/IP needs to be configured before the HomeRun server will function. Before continuing, make sure your Windows PC client can *ping* the System i system. See one of the following IBM System i references for information on configuring TCP/IP.

- *TCP/IP Fastpath Setup (SC41-3430)*
- *TCP/IP Configuration and Reference (SC41-3420)*

✓ The installation uses TCP/IP port number **{default port}**. If that port is in use, the installation will search for unused port numbers starting at **{default port}** and continue looking until one is found. To determine the port that was chosen use the following command:

```
WRKSRVTBLE SERVICE(CENTERFIELD_SERVER_80)
```

✓ The HomeRun server installs into a library named *{PROGRAM LIBRARY}. The installation process will copy over existing programs if the library exists.*

✓ The HomeRun data files install into a library named *{DATA LIBRARY}. The installation process will selectively replace the data files and preserve information specific to your configuration and settings if they exist*.

✓ The HomeRun server requires an active subsystem for proper installation and operation. Determine the subsystem you wish to use. It will be needed later in the installation process. If you have no special work management requirements, you may use the default of *{PROGRAM*

LIBRARY}/XCSBS80.

✓ If you are installing over the top of an existing HomeRun or Database Essentials product library, the jobs currently using that library will be ended before the installation begins. The server will be re-started at the end of the installation if a valid password is entered during the install, or if you have installed over an existing library that already has a valid password.

2) Sign-on to a 5250 session with QSECOFR or a user profile that has security officer special authorities. This user profile must have a minimum of *ALLOBJ, *IOSYSCFG, *JOBCTL, *SECADM, and *SAVSYS special authorities.

3) The system value QUSEADPAUT must be set to *NONE.

You can verify and change this system value using the following command.

```
WRKSYSVAL QUSEADPAUT
```

In addition, if the system value QUSEADPAUT is secured with an authorization list, the user profile used for the installation must be on that authorization list.

4) Place the CD-ROM containing the HomeRun product into your System i primary CD-ROM drive and issue the following command (does not apply when performing a fully electronic installation).

```
LODRUN *OPT
```

5) When you are prompted to configure the HomeRun server subsystem, either take the default (i.e. XCSBS80) or enter the subsystem where you want the HomeRun server and client jobs to run, and press Enter. It is recommended that the default be taken. You can reconfigure the server's subsystem after the installation by running the CFGSVR command.

6) When you are prompted to enter the HomeRun license password, enter the license password that was provided with your product and press Enter. The license password is based on the serial number of the System i and the LPAR number. Hence, a different license password is needed for each System i that you install.

If you do not know the license password for your System i, you can skip this step of the installation by pressing F12. You can enter the license password after the installation by running the PASSWORD command from the *{PROGRAM LIBRARY}* library.

If you defer adding the license password, the HomeRun server will not start automatically. No client machines will be able to connect to the System i using any of the HomeRun tools until the password is entered.

7) You will be returned to an System i command line when the installation is complete. If you did not skip any of the installation steps, the HomeRun server will be automatically started for you. If you skipped any steps in the installation process, you should perform the configuration steps that you skipped before you continue.

8) If you skipped configuring the server or the license password as part of the installation, or if the server did not automatically start as part of the product installation, you need to start the HomeRun server before clients can connect to the System i using any of the HomeRun tools. Prior to trying to start the server, you should ensure that TCP/IP is active on your system. If it is not active, issue the following command using the correct options for starting the TCP/IP servers on your system.

   STRTCP STRSVR(***YES/*NO**)

9) Issue the following command to start the HomeRun server:

   {PROGRAM LIBRARY}/STRSVR

10) Verify the server is active in the subsystem that you specified. The job name will be XCSERVER. The following command can be used to check for the active job and to verify that the subsystem is correct.

   WRKACTJOB JOB(XCSERVER*)

11) The server will need to be restarted every time the associated subsystem is ended or the system is restarted. You may add the *{PROGRAM LIBRARY}*/STRSVR command to your system startup routine **(recommended)** or add an autostart job entry in the subsystem if you would like to automatically start the server. If you start the server in your system startup procedure, the i5/OS command STRTCP must be issued and TCP must be completely started before the server will start.

12) If you plan to use insure/SECURITY or insure/MONITOR, you will need to register Centerfield's exit point programs with the System i registration facility:

   {PROGRAM LIBRARY}/ADDMON

   This enables the policies you define to take effect. More information can be

found in the manuals for each of these tools.
NOTE: you will have to recycle the host servers for the exit point registration to take effect (refer to the table on next page as well as the Appendix for more detailed information).

13) If you plan to use insure/RESOURCES, you must install policy managers before your resource policies will take effect. This can be done either with the:

```
{PROGRAM LIBRARY}/ADDQCEXIT
```

or with the menus in insure/RESOURCES PC client GUI.

ODBC exit may need recycling database host server and corresponding prestart jobs (i.e. ENDHOSTSVR *DATABASE, ENDPJ QUSRWRK QZDASOINIT, STRHOSTSVR *DATABASE)
More information can be found in the insure/RESOURCES manual.

14) You are now ready to begin the workstation installation.

3.1.2.2 Windows 2000/XP/Vista/7 Installation

We recommend Administrator authority on the PC to run the install:

1) Go to wedsite: http://www.centerfieldtechnology.com/downloads.asp

2) Click on the download link for HomeRun

3) Follow the online instructions in the START HERE pdf.

4) If the installation needed to replace DLLs, you will be prompted to restart your system.

5) You may now use the HomeRun tools for which you have a license.

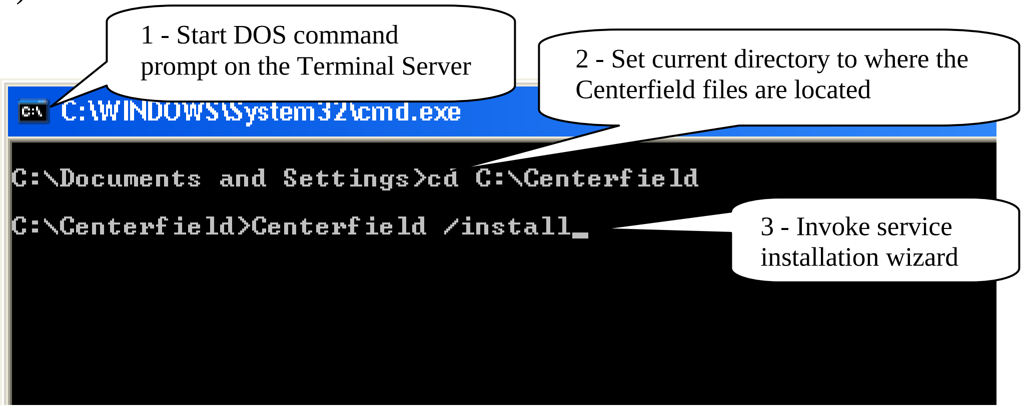
3.1.2.3 Installation instructions for OneWorld® customers using Windows® Terminal Server (i.e. Citrix® server)

These instruction apply **only** to customers that are running JDE OneWorld® (a.k.a. PeopleSoft EnterpriseOne® aka Oracle E1®) on their System i and have purchased insure/Monitor for OneWorld® license. Furthermore, customers using Terminal Servers (TS) in their environment and require Centerfield windows service running on those TS to achieve correlation of System i ODBC jobs to the OneWorld users will follow these instructions.

If these conditions do **NOT** apply to your scenario, do **NOT** follow these installation steps.

These instructions are to be performed by the Terminal Server administrator (i.e. OneWorld® CNC specialist) with Administrator authority to the server.

1) Locate **Centerfield.exe, Centerfield.dll and CenterfieldServiceConfiguration.exe** on the installation CD using Windows Explorer
2) Create a new folder on the Terminal Server (i.e. C:\Centerfield)
3) Copy and paste **Centerfield.exe, Centerfield.dll and CenterfieldServiceConfiguration.exe** to the newly created folder
4)



1 - Start DOS command prompt on the Terminal Server

2 - Set current directory to where the Centerfield files are located

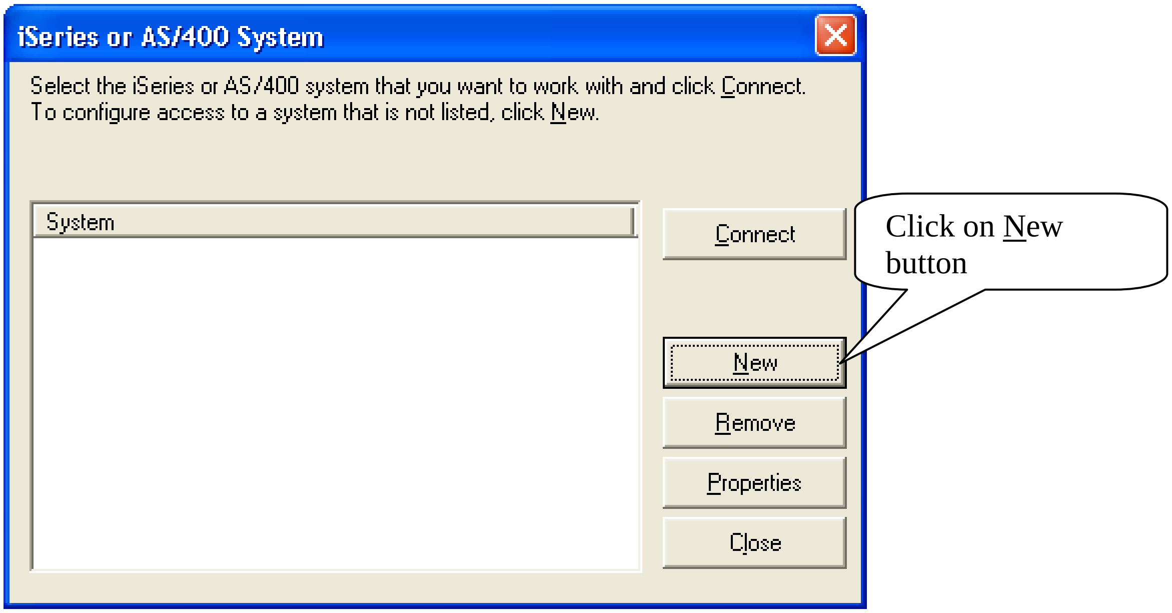
3 - Invoke service installation wizard

```
C:\WINDOWS\System32\cmd.exe

C:\Documents and Settings>cd C:\Centerfield

C:\Centerfield>Centerfield /install_
```

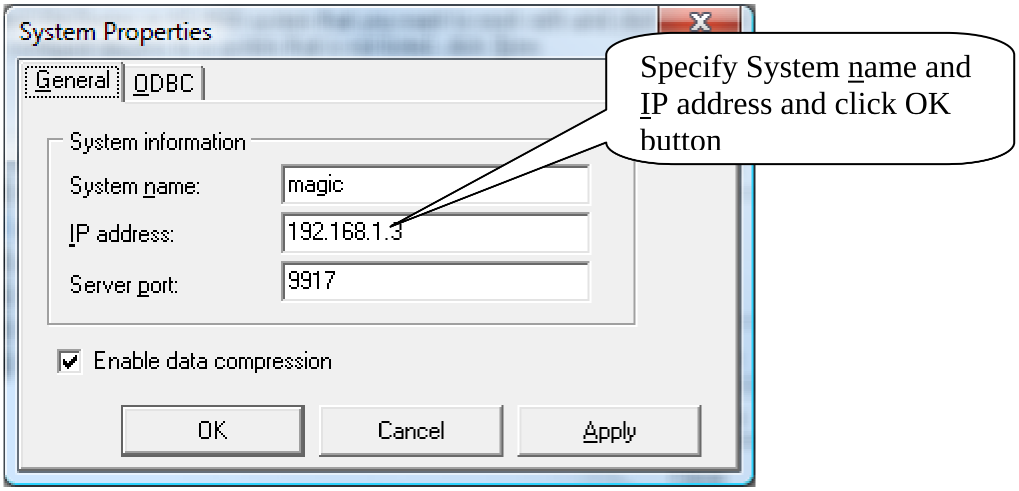
5) Several one time configuration windows will appear.



iSeries or AS/400 System

Select the iSeries or AS/400 system that you want to work with and click Connect.
To configure access to a system that is not listed, click New.

Click on New button



System Properties

System information

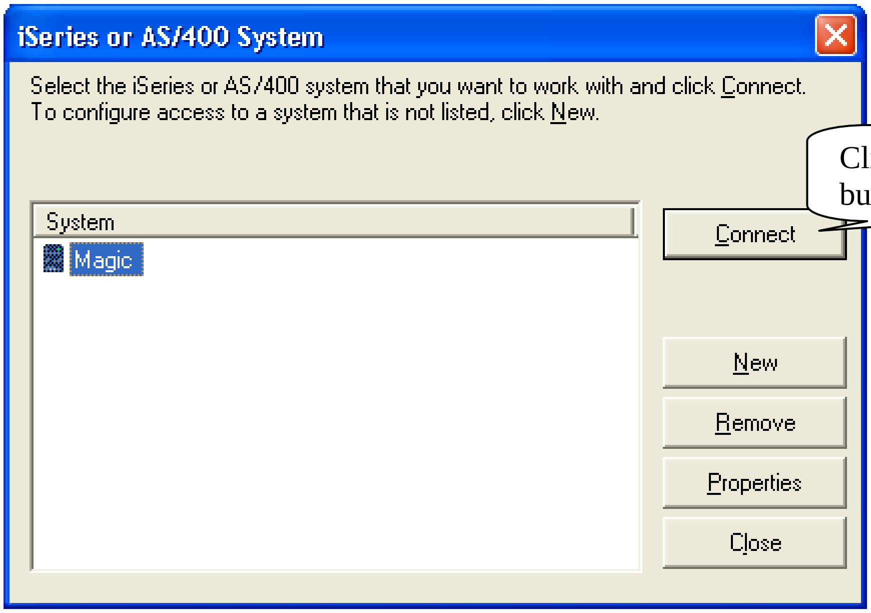
System name: magic

IP address: 192.168.1.3

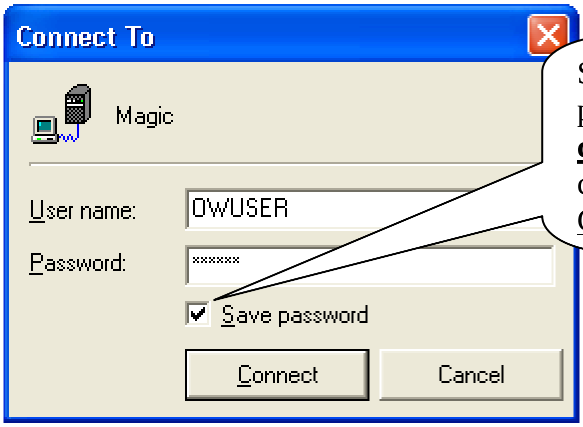
Server port: 9917

☑ Enable data compression

Specify System name and IP address and click OK button


iSeries or AS/400 System
Select the iSeries or AS/400 system that you want to work with and click Connect.
To configure access to a system that is not listed, click New.
System
Magic
Connect
New
Remove
Properties
Close

Click on Connect button

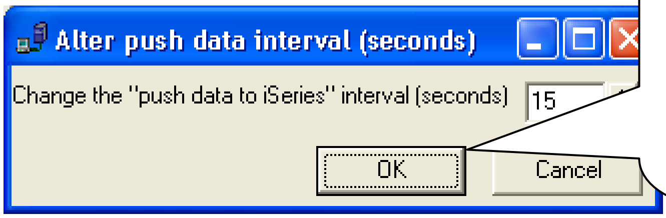


User name: OWUSER

Password: ******

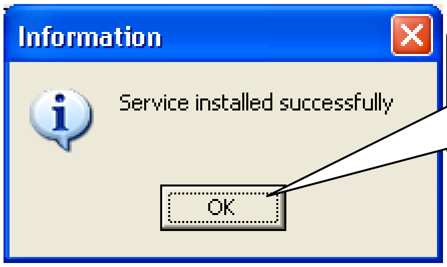
☑ Save password

Specify iSeries profile and password, make sure you **check** the Save password checkbox then hit the Connect button.



Change the "push data to iSeries" interval (seconds) 15

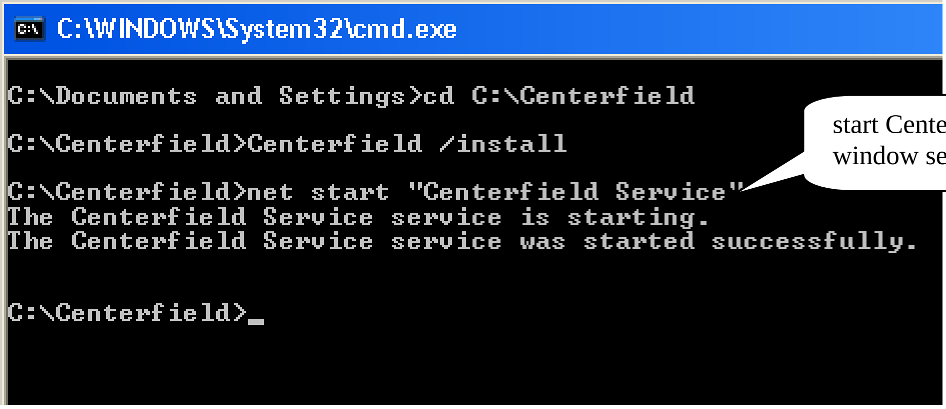
Default 'push data to System i' interval is 15 seconds, but you can change it at this time.
Good rule of thumb is to use 5 seconds per Terminal Server on which our service is running (i.e. 8 * 5 = 40 seconds).
When done click OK button



Service installed successfully

Upon successful installation of the Centerfield Terminal Server windows service you will see this window. Click OK to return to DOS prompt screen.

6)



```
C:\Documents and Settings>cd C:\Centerfield

C:\Centerfield>Centerfield /install

C:\Centerfield>net start "Centerfield Service"
The Centerfield Service service is starting.
The Centerfield Service service was started successfully.


C:\Centerfield>_
```

start Centerfield window service

You can check Centerfield service status in a couple of ways. One is to invoke the services applet via Start->Run->services.msc. Another is to check for the existence of Centerfield.exe process in the Task Manager on the Terminal Server.

If there are active OneWorld connections on the Terminal Server, there should be an XCCLIENT job on the System i with the joblog message:
“Serving Terminal Server at IP address 0.0.0.0.” where 0.0.0.0 will list the true IP address of the Terminal Server. If there are no active OneWorld connections on the Terminal Server, Centerfield service will not push any data to the System i.

This completes the Centerfield service installation on the Terminal Server. Service should auto-start every time Terminal Server reboots. For the Centerfield service to push data, XCSERVER job needs to be active on the System i. You can start it by executing *{PROGRAM LIBRARY}*/STRSVR command, or automate this by adding STRSVR command to your IPL startup routine (DSPSYSVAL QSTRUPPGM).

There is one final configuration step that has to be performed from the insure/MONITOR for OneWorld® PC client.
Refer to the documentation located under section **“Configuration for OneWorld® customers”** in the **“insure Monitor User Guide and Reference”** document for details.

3.1.3 Connecting to the server for the first time

Once you have the product installed, you are ready to connect to the server from the Windows based workstations. The workstation installation adds a program group to your start menu called *Centerfield HomeRun*. Within the program group there are one more features that can be selected depending on the options that were selected at installation time. To start the HomeRun tool you want to use, choose its name from this group.

The HomeRun tools use a consistent interface for connecting to the System i. When you start one of the HomeRun tools for the first time on your workstation, you will need to configure your connection to a server.

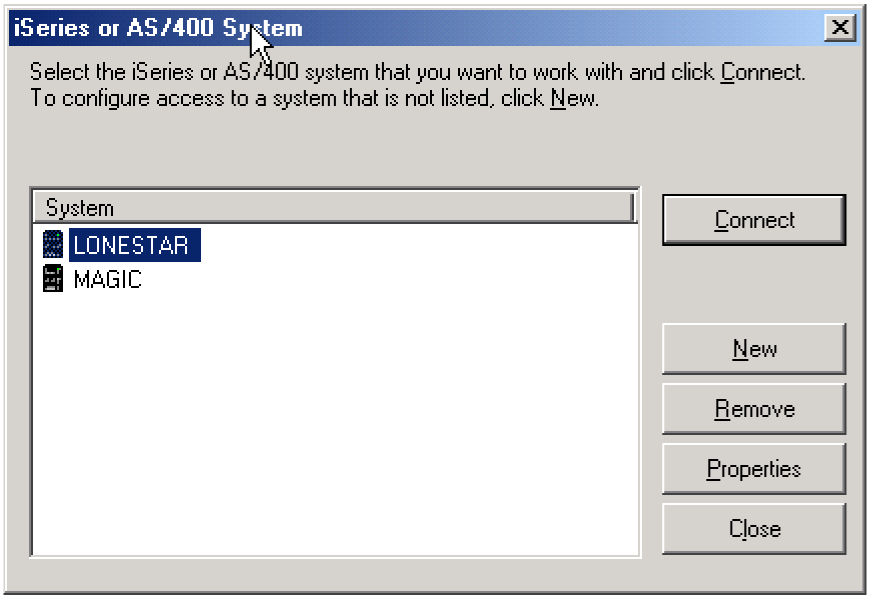
Auto-configuration

When you start the tool, it will try to auto-detect the System i systems in your environment and configure itself to connect to them.

The auto-configuration interface will show you all of the systems that it can detect. If no systems are detected, the auto-configuration will be skipped and you will need to define your system connection information manually.

Manual configuration

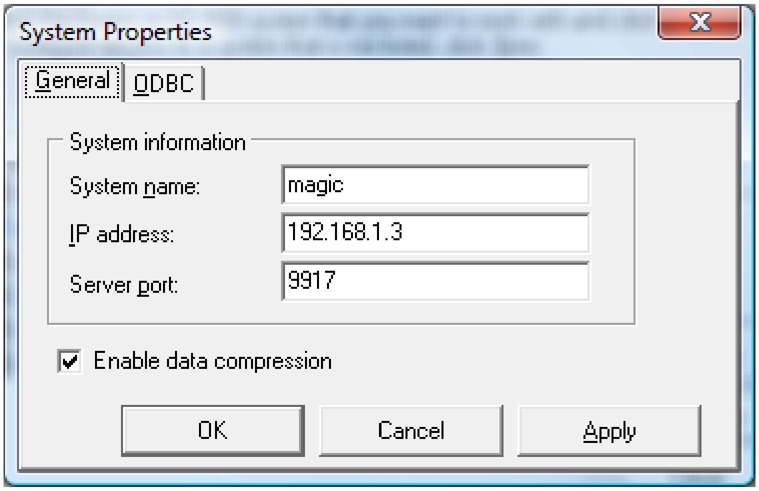
If the auto-configuration interface did not detect your system, or you want to manually configure a connection, use the following instructions.



iSeries or AS/400 System

Select the iSeries or AS/400 system that you want to work with and click Connect.
To configure access to a system that is not listed, click New.

When the *System i system* window is displayed, press **[New]** to add a new system.



System Properties

System information

System name: magic

IP address: 192.168.1.3

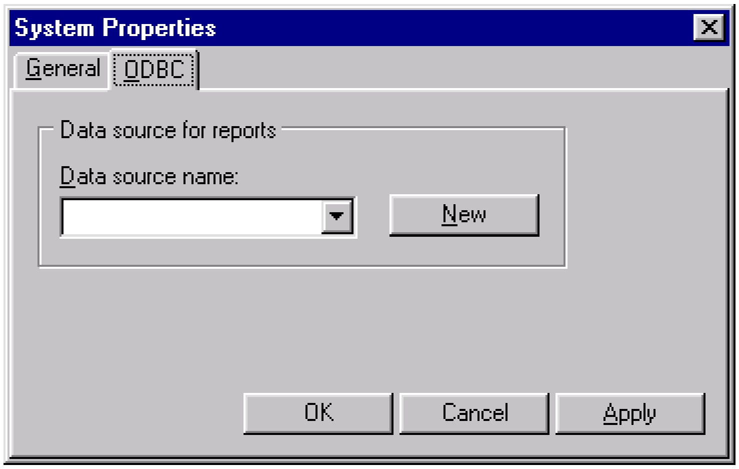
Server port: 9917

☑ Enable data compression

For *System name*, enter a descriptive name that identifies your system.

For *IP address*, enter either the symbolic name or dotted IP address for your System i system.

For the *TCP/IP port*, use the port that was selected during the System i installation process, by default **{default port}** is used.

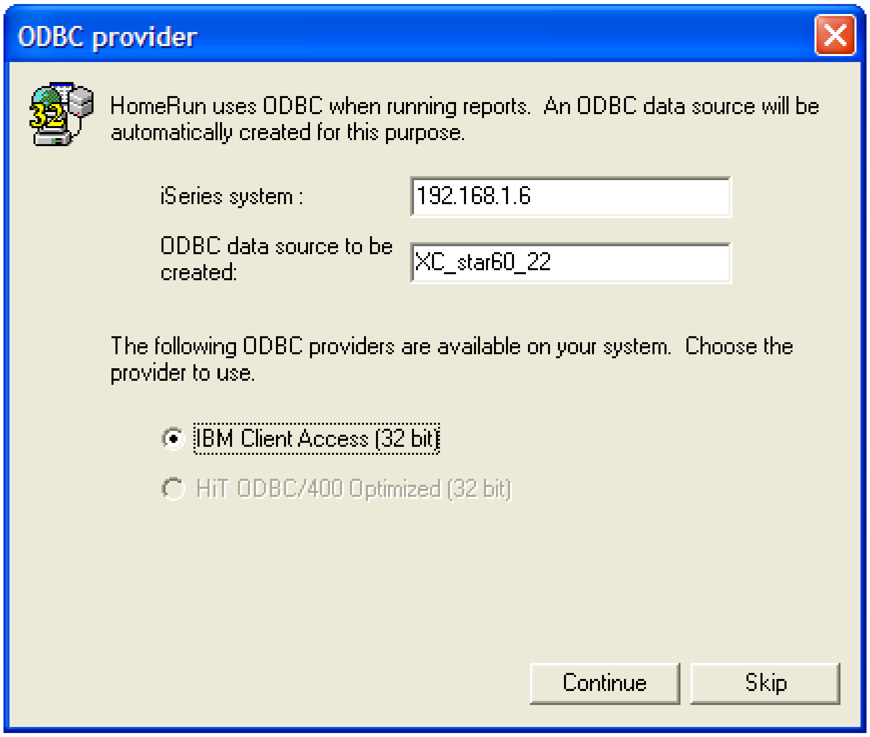


System Properties

Data source for reports

Data source name:

On the *ODBC* tab, specify a name for your ODBC data source and press **[New]** to create a data source for this system. If you already have a Client Access ODBC data source that is configured to your system, you can specify your existing data source. If you choose to create a data source, you will see a display similar to the following.



HomeRun uses ODBC when running reports. An ODBC data source will be automatically created for this purpose.

iSeries system : 192.168.1.6

ODBC data source to be created: XC_star60_22

The following ODBC providers are available on your system. Choose the provider to use.

● IBM Client Access (32 bit)

○ HiT ODBC/400 Optimized (32 bit)

Specify the name of the System i as defined in your connection software, and select the data source type that you want, and press **[Continue]**. You must have either Client Access or HIT ODBC installed to use this fast path interface for creating your data source. If you have another ODBC provider, you must create the data source with the ODBC Administrator utility that is provided with Windows and then specify the data source name within the *System Properties* window on the *ODBC* tab.

After you have created your data source and specified values for all of the appropriate prompts on the *System properties* window, press **[OK]** to create the connection definition.

Connecting to a system

Select the system that you want to work with from the list provided in the *System i system* window, and press the **[Connect]** button. Enter your user profile and password and press **[Connect]**.

3.2 Using ADDMON

Before the policies you define in insure/SECURITY will take effect, you must run the Centerfield command ADDMON, found in the *{PROGRAM LIBRARY}* installation library.

This command tells the System i server which remote interface you are going to secure by configuring the proper Centerfield programs at various IBM exit points. You do not need to understand exit points to use this support; however, more detailed information is available in the Appendix entitled *Enabling Remote Monitoring*.

To enable insure/SECURITY to secure all covered interfaces by default, specify *{PROGRAM LIBRARY}*/ADDMON DRVTYPE(*ALL). This is the recommended default. If your environment requires that you pick and choose interfaces to which to add exit points, prompt on the ADDMON command and choose the interfaces.

The following table shows how the options on the ADDMON DRVTYPE parameter map to the options available for you to secure in the insure/SECURITY interface:

ADDMON parameter	**Interface Description**
*CSCM	Central Server client manager
*CSCONV	Central Server conversion map
*CSLM	Central Server license manager
*DTAQ	Client Access - Data queue server
*DTAQORIG	Client Access - Original data queue server
*FILETRANS	Client Access - Original file transfer function
*FILSRV	File server

*IBMDDM	Distributed Data Management DRDA - FileTek products DRDA - Grandview DB/DC Systems products DRDA - IBM DB2 Connect (formally DDCS) DRDA - IBM DB2 for VSE and VM DRDA - IBM DB2 UDB for System i DRDA - IBM DB2 UDB for OS/390 DRDA - Informix Software products DRDA - Oracle Corporation products DRDA - StarQuest products DRDA - Wall Data Rumba for Database Access DRDA - XDB Systems products Hit ODBC for DB2 DataDirect Technologies Derby Network Client SeeBeyond ICAN (Sun JCAPS) Java Client For more information on DRDA client support and product identifiers, visit the website: http://www.opengroup.org/dbiop/uploads/40/9358/drda-pid.htm
*IBMFTP	IBM File Transfer Protocol (FTP) Client IBM File Transfer Protocol (FTP) Server IBM REXX File Transfer Protocol (FTP) Server IBM Trivial File Transfer Protocol (FTP) Server
*IBMODBC	IBM Client Access ODBC and Hit Optimized ODBC
*LMORIG	Client Access - Original License Manager server
*MSGORIG	Client Access - Original message server
*NETPRT	Client Access - Network print server - entry
*REXEC	REXEC server logon
*RMTCALL	Client Access - program call
*RMTCMD	Client Access - remote command
*RMTSQL	Client Access - Original Remote SQL server
*SOCKSRV	TCP signon server
*TELNET	Telnet
*VRTPRT	Client Access - Original virtual print server

The time at which a new exit program is recognized by the System i server varies from one exit point to another. Some take effect immediately; others, such as exit points on the Client Access host servers, do not take effect until the host server is next started. Still others, including the File Server, require that the subsystem (QSERVER in this case) be

ended and restarted before the new program takes effect. If you find that insure/SECURITY is not having the effect you expected, end the server programs and restart them or, if possible in your environment, end the subsystem and restart it.
There are some exit point jobs (i.e. QTVDEVICE telnet server jobs) that persist even after the host server is recycled. These jobs may need explicit end and restart to become aware of the newly registered exit point program.
NOTE: IPL or bringing system to restricted state should recycle all exit point interfaces unconditionally.

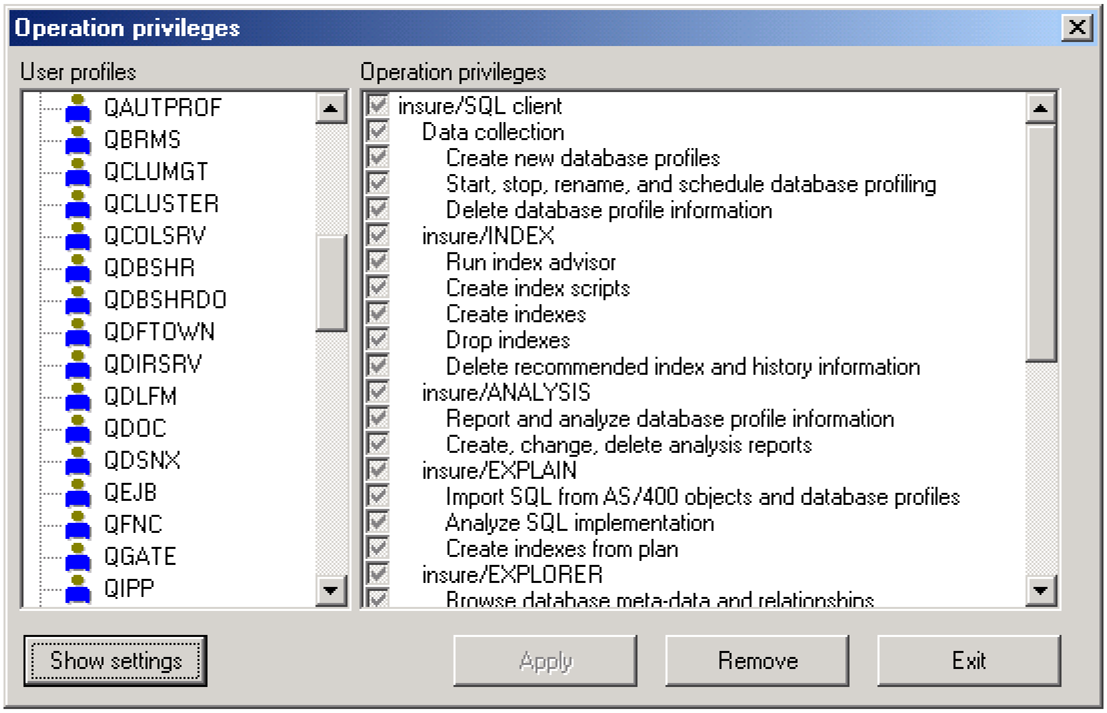
3.3 Securing the use of HomeRun tools

HomeRun provides a broad suite of powerful functionality to the IT organization. Often, duties are split among various specialists. In this situation, it may be desirable to control what functions in the toolset each specialist can use. HomeRun provides a set of granular controls to allow an IT administrator to determine who can and should use each part of the toolset.

To implement security settings for a user or group profile use the following procedure:

1. In any of the HomeRun tools, select Tools | Privileges.

2. The next dialog has a list of users and groups on the left side and an itemized list of product features on the right as seen below:



QAUTPROF
QBRMS
QCLUMGT
QCLUSTER
QCOLSRV
QDBSHR
QDBSHRDO
QDFTOWN
QDIRSRV
QDLFM
QDOC
QDSNX
QEJB
QFNC
QGATE
QIPP

- ☑ insure/SQL client
  - ☑ Data collection
    - ☑ Create new database profiles
    - ☑ Start, stop, rename, and schedule database profiling
    - ☑ Delete database profile information
  - ☑ insure/INDEX
    - ☑ Run index advisor
    - ☑ Create index scripts
    - ☑ Create indexes
    - ☑ Drop indexes
    - ☑ Delete recommended index and history information
  - ☑ insure/ANALYSIS
    - ☑ Report and analyze database profile information
    - ☑ Create, change, delete analysis reports
  - ☑ insure/EXPLAIN
    - ☑ Import SQL from AS/400 objects and database profiles
    - ☑ Analyze SQL implementation
    - ☑ Create indexes from plan
  - ☑ insure/EXPLORER
    - ☑ Browse database meta-data and relationships

3. To quickly restrict usage to a few users, select the *PUBLIC user profile and unclick the top-most box. At the same time, you should make sure one user profile has the capability to update privileges. To do that, pick one user profile and make sure *Administration* and *Edit HomeRun privileges* are checked. If you do not do this step, only the QSECOFR profile will still be able to edit privileges.

4. To change the settings for a particular user, click on their profile and push the “Show

settings" button. On the right you will then see the parts of the product that profile is authorized to use. If the box is gray, the user does not have explicit authority defined and will get authority from either their group profile or global settings (the authority set for *PUBLIC).

5. To restrict a user from a particular feature, simply uncheck the box in front of that feature and click the "Apply" button. If the user attempts to use that function, they will be given a message saying they are not authorized to that feature and that they should contact their system administrator.

6. To remove settings, choose a profile (or set of profiles with multi-select) and click "Remove settings". This will result in all settings being picked up from the group profile or *PUBLIC.

NOTE: When settings are changed for a user, they will not take effect until the user connects to that iSeries again (i.e. exits the client GUI and restarts it).

3.4 Security issues with remote interfaces

Before implementing any security with insure/SECURITY, you should decide whether you need to address the issue of data visibility, system access, or both. If you need to restrict users' access to some data when they are connected with a remote interface, you will use insure/SECURITY's proxy user feature. If you need to restrict users from using a given remote interface, or restrict the times during which the interface can be used, you will use insure/Security policy feature. This feature allows you to define usage policies for a single user or a group of users.

3.4.1 Data visibility

In many System i shops, it is common to have applications that do not use the System i object-level security, or do not use it consistently. These applications protect sensitive data through several methods:

- Simply not showing data on a user's screen
- Restricting the use of menus or menu options based on user profiles
- Allowing or disallowing data to be updated based on application logic
- Only allowing access to certain queries

These methods work well in an environment where users can be restricted from using a System i command line and are forced into a pre-defined application when they sign on. Unfortunately, these methods do not work when users connect to your System i over a network interface that bypasses these protection mechanisms.

One of these interfaces, Open Database Connectivity (ODBC) is probably the most popular way to access System i data from a Windows PC. Many easy-to-use tools are available which allow even novice computer users to access data. Several issues surface when this happens:

- Data previously hidden and unavailable to users now can be seen
- Data previously seen can now be updated or deleted
- Entire files can now be deleted with a single command from the PC (even if the user does not have *OBJEXIST rights to the file)

One approach to these problems is to define System i object level security for the data you need to protect. However, this has several ramifications which may not be acceptable in your environment including:

- Existing applications may fail to work because they depend on the currently-defined object security.

- You may need to have different levels of security based on the application (some may need to only read the data while others need update authority).
- You need to define data-specific security (e.g. sales managers from the west should only see data from the west region).

insure/Security lets you define object-level security for users when they use the system via remote interfaces *without* interfering with the way applications work when users sign on to a green screen. It does this through the concept of ***proxy users***, which will be explained below.

3.4.2 System Access

The availability of remote interfaces creates another concern for IT organizations: controlling access to the System i by unauthorized users. IT often has the following goals:

- Control which data access methods are employed by user (e.g. only let certain people use FTP)
- Restrict the use of certain products by time of day (e.g. only allow ODBC access during normal working hours)
- Standardize on software so they do not have to learn and support a large number of products.

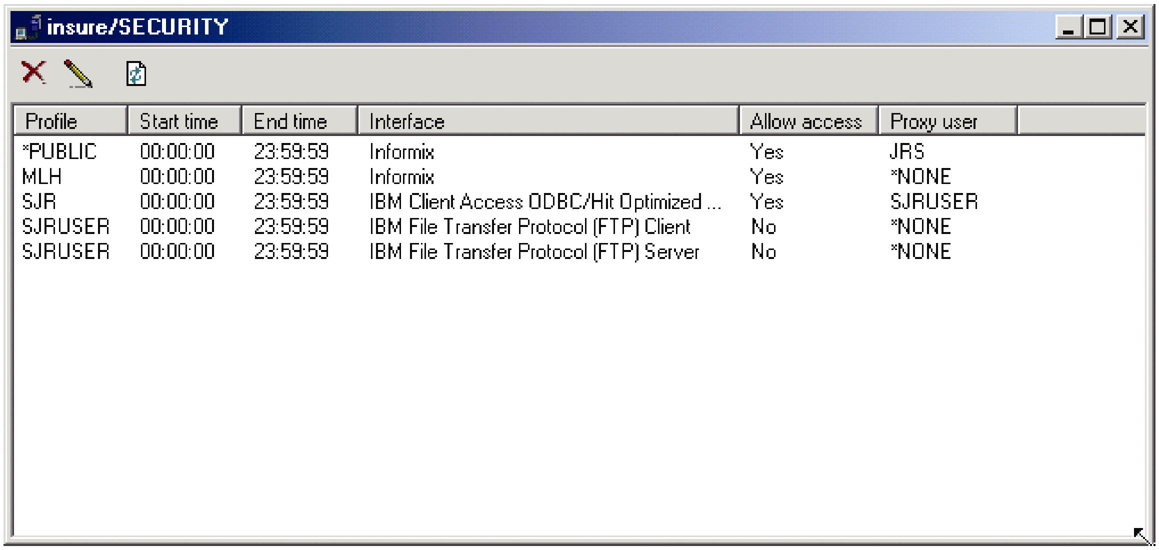
The insure/SECURITY tool lets you implement security to meet these objectives with very little effort.

3.5 Basic process to set up insure/Security

The process used to control the use of data access interfaces depends on the objective you want to accomplish. The following sections describe the steps required to implement various types of control.

3.5.1 Controlling use of a remote interface globally:

1. Start insure/Security by choosing *Security* from the Welcome dialog. Alternatively, you can use the menu item *Tools | insure/Security*. You will see a dialog similar to the one below.



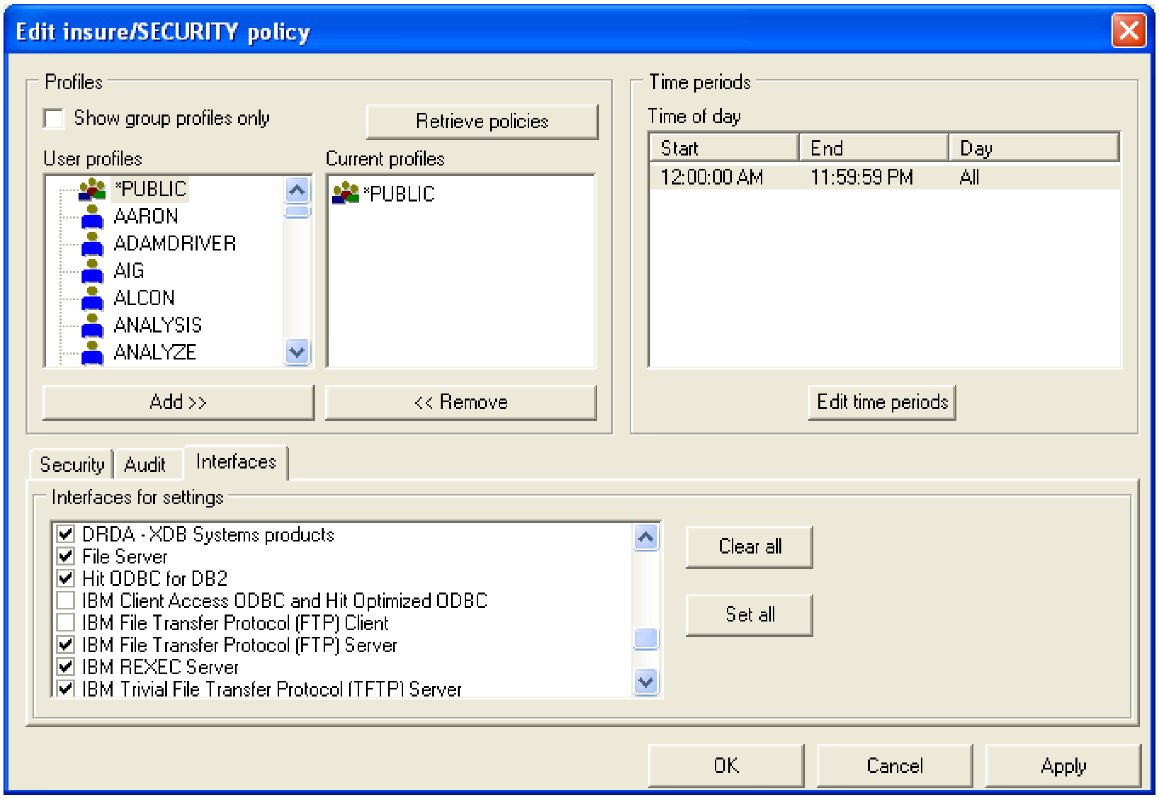
Profile	Start time	End time	Interface	Allow access	Proxy user
*PUBLIC	00:00:00	23:59:59	Informix	Yes	JRS
MLH	00:00:00	23:59:59	Informix	Yes	*NONE
SJR	00:00:00	23:59:59	IBM Client Access ODBC/Hit Optimized ...	Yes	SJRUSER
SJRUSER	00:00:00	23:59:59	IBM File Transfer Protocol (FTP) Client	No	*NONE
SJRUSER	00:00:00	23:59:59	IBM File Transfer Protocol (FTP) Server	No	*NONE

2. Select *Security|Start Policy Editor* from the menu bar.

3. Choose *PUBLIC as the user profile to alter and click **[Add>>]**.

4. Choose the *Exclude access to the system* option.

5. Click the *Interfaces* tab and ensure all of the interfaces you want to restrict have a check mark in the box in front of the description. Once you have completed you should see a window similar to the following. In this case, all users of the system are restricted from all interfaces except IBM Client/Access ODBC and the FTP client. When you are finished, click the **[Apply]** button to use these settings.



□ Show group profiles only

User profiles: *PUBLIC, AARON, ADAMDRIVER, AIG, ALCON, ANALYSIS, ANALYZE

Current profiles: *PUBLIC

Time of day

Start	End	Day
12:00:00 AM	11:59:59 PM	All

Interfaces for settings

☑ DRDA - XDB Systems products
☑ File Server
☑ Hit ODBC for DB2
□ IBM Client Access ODBC and Hit Optimized ODBC
□ IBM File Transfer Protocol (FTP) Client
☑ IBM File Transfer Protocol (FTP) Server
☑ IBM REXEC Server
☑ IBM Trivial File Transfer Protocol (TFTP) Server

3.5.2 Interfaces supported by insure/SECURITY

Supported interfaces include those described in the following four tables:

Client Access Original Servers

The original servers were used by Client Access for DOS, Client Access for DOS with Extended Memory, and Client Access for OS/2 Functions. You can monitor or secure these exit points to ensure any clients using these host servers conform to your environment.

Note: HomeRun implements its monitoring and security for these servers via exit points registered in the registration facility. To have the system check the registration facility for these programs, you must set the PCSACC network attribute to *REGFAC. See IBM's documentation for the Change Network Attributes (CHGNETA) command for help if you need to change this attribute.

insure/RESOURCES and insure/SECURITY name	Description	Enforced when ADDMON parameter is used:
Client Access – Original virtual print server	Used by original Client Access clients' Virtual Print function. Allows use of a printer that is attached to the host system as though the printer was directly attached to a personal computer.	*VRTPRT, plus PCSACC = *REGFAC
Client Access – Original data queue server	Original Client Access provides APIs that use this server to allow PC applications to work with System i data queues.	*DTAQORIG, plus PCSACC = *REGFAC
Client Access – Original file transfer function	PC Support File Transfer Function and HIT ODBC for file transfer. It also refers to Client Access File Transfer Function for Version 3 Release 1 Modification 3 or early versions.	*FILETRANS, plus PCSACC = *REGFAC

Client Access – Original message server	Original message function routes messages that are sent from PC users to the appropriate user and receives messages for PC users and sends them to the PC workstation.	*LMORIG, plus PCSACC = *REGFAC
Client Access – Original License Manager server	Original license server ensured valid licenses for original Client Access clients.	*MSGORIG, plus PCSACC = *REGFAC
Client Access – Original Remote SQL server	Original Client Access provides APIs that use this server to allow PC applications to run SQL statements on a System i.	*RMTSQL, plus PCSACC = *REGFAC

Client Access Host Servers

These servers are used by Client Access Express, and may be used by other products. These are the servers which are started by the STRHOSTSVR command.

insure/RESOURCES and insure/SECURITY name	Description	Enforced when ADDMON parameter is used:
Central Server client manager	Runs an exit program for all client management requests received by the central server.	*CSCM
Central Server conversion map	Retrieves conversion maps for PC applications which require them when they connect to the System i.	*CSCONV
Central Server license manager	Used for license management requests, including those from Client Access.	*CSLM
Client Access – program call	Allows client applications to call System i programs and pass parameters.	*RMTCALL
Client Access – remote command	Allows client users and applications to issue System i CL commands.	*RMTCMD
Client Access – Data queue server	APIs that can allow PC applications to work with System i data queues.	*DTAQ
Client Access – Network print server – entry	Allows enhanced client control over print resources on the System i server.	*NETPRT
File server	Allows clients to store and access information, such as files and programs, located on the System i server.	*FILESRV
TCP signon server	Runs when various signon requests are received, including Retrieve signon information, Change password , and Generate authentication token	*SOCKSRV

Other Registered Exit Points

These interfaces are monitored and secured by HomeRun via exit programs registered with the registration facility.

insure/RESOURCES and insure/SECURITY name	Description	Enforced when ADDMON parameter is used:
IBM File Transfer Protocol (FTP) Client	Controls access to the System i FTP client at validation time.	*IBMFTP
IBM File Transfer Protocol (FTP) Server	Controls access to the FTP server at validation time.	*IBMFTP
IBM REXEC Server	Controls access to Remote Execution Server at validation time.	*IBMFTP
IBM Trivial File Transfer Protocol (TFTP) Server	Trivial file transfer protocol (TFTP) provides basic file transfer with no user authentication. This protocol provides support for the IBM Network Station for System i.	*IBMFTP
IBM Client Access ODBC and Hit Optimized ODBC	The Client Access ODBC driver, HIT ODBC based on the Optimized Database Server, Client Access File Transfer Function, and other interfaces that use the Optimized Database Server.	*IBMODBC
REXEC server logon	Controls the authentication of users to a TCP/IP application server at logon time.	*REXEC
Telnet	Hooks into the Telnet interface's signon logic.	*TELNET

DDM Server and Other interfaces

These interfaces are monitored and secured by HomeRun via the DDMACC network attribute.

insure/RESOURCES and insure/SECURITY name	Description	Enforced when ADDMON parameter is used:
Distributed Data Management	IBM DDM	*IBMDDM
DRDA – FileTek products	FileTek products which use the DRDA interface	*IBMDDM
DRDA – Grandview DB/DC Systems products	Grandview products which use the DRDA interface	*IBMDDM
DRDA – Informix Software products	Informix products which use the DRDA interface	*IBMDDM
DRDA – IBM DB2 for VSE and VM	Connections from IBM DB2 for VSE and VM which use the DRDA interface	*IBMDDM
DRDA – IBM DB2 Connect (formerly DDCS)	Connections from DB2 Connect	*IBMDDM
DRDA – IBM DB2 UDB for System i	Connections from IBM DB2 UDB for System i which use the DRDA interface	*IBMDDM
DRDA – IBM DB2 UDB for OS/390	Connections from IBM DB2 UDB for OS/390 which use the DRDA interface	*IBMDDM
DRDA – Oracle Corporation products	Oracle products which use the DRDA interface	*IBMDDM
DRDA – StarQuest products	StarQuest's ODBC and DRDA products including StartSQL, StarPipes, and Host Data Replicator. If you are using the ODBC driver that is shipped as part of Windows BackOffice SNA server or other Microsoft product, StarQuest is most likely the manufacturer.	*IBMDDM

DRDA – Wall Data Rumba for Database Access	Wall Data products which use the DRDA interface	*IBMDDM
DRDA – XDB Systems products	XDB Systems products which use the DRDA interface	*IBMDDM
HIT ODBC for DB2	HIT Software ODBC products that are DRDA based	*IBMDDM

3.5.3 Restricting use of a remote interface by time of day

1. Select the user, set of users, or group profile you want to configure settings for.
2. Choose *Exclude access to the system.*
3. Edit the time period shown in the upper right of the window.
4. Click the *Interfaces* tab and choose which interfaces should be restricted by time of day.

3.5.4 Managing use of a remote interface by individual user

If you either want to allow or restrict the use of a interface by user, use the following steps:

1. Select the user, set of users, or group profile you want to configure settings for.
2. Choose *Allow Connection* or *Exclude access to the system* depending on your purpose.
3. Click the *Interfaces* tab and choose which interfaces should be allowed and/or restricted

3.5.5 Creating proxy users for remote interfaces

You will want to use this method when you want to restrict users' access to certain data when they connect with a remote interface, yet leave their current authorities in place when they sign on to a green screen. The basic process for setting up proxy users is:

1. Identify security requirements
2. Identify classes of users
3. Create proxy user profiles and assign the appropriate authorities

4. Assign users or groups to the appropriate proxy user
5. Test to ensure the proper authority is available

As a general rule, the suggested approach to adding security to your system will be to lock everything down and to selectively let users have access to the objects they need. This approach ensures that you do not miss important objects.

The following sections will take you through each of these steps and explain different options to accomplish that task.

3.5.5.1 Step One: Identifying security requirements

The first step in closing security holes is to determine what data should be made available to end-users that use ODBC. In general, you should already have an idea of what data should and should not be made available. In environments where users simply use a query tool to generate reports on their Windows desktop, it should be relatively easy to identify which files are accessed through one of the following methods:

- If you are licensed to insure/ANALYSIS, profile database activity and run reports that identify which files are accessed
- Survey the users
- Look at the queries being submitted with the tool used to run them

For ODBC applications that update data similar approaches can be taken. In addition, you can either look at the application code or run ODBC traces to determine the data needs of each transaction type.

To determine what authorities are currently enforced at the object level you can use the DSPOBJAUT command and redirect the output to a file. That file can then used to create reports which identify mismatches between the real security requirements and the current state of the system.

3.5.5.2 Step Two: Identifying classes of users

The next step is to identify what types of users need to be created. In many situations it will probably be adequate to define a class with no authority to most objects and read-only authority to the rest. Different flavors of this “read-only” class can be defined depending on the need to authorize access to certain data. For example, you may have a read-only class for your marketing department that allows all members of that department to run queries against the sales history database. The human resources department, on the other hand, may only need access to the personnel data and not the sales information. In this case there would be two classes, both having read access to their respective data.

3.5.5.3 Step Three: Creating user classes and assigning

authorities

Class creation: The next step is to actually create the classes. A “proxy” user profile will be used to implement the classes defined in the second step. To accomplish this, you will create a new user profile on the System i and associate existing profiles with this “proxy”. Each user will be associated with the proxy profile so its authorities (rather than the actual user’s authorities) are used by the System i when data is accessed via ODBC.

Object authority assignment: Once the proxy user profiles are created, you need to assign the appropriate authority to them based on the needs of their class. To minimize the amount of work required to define the appropriate authority, it is best to start at the library level and define *EXCLUDE authority for libraries that should not be accessed at all. For libraries that users should be able to access you will then need to define the needed authority at the object level. Taking the “lock down approach”, you would define *EXCLUDE authority for all of the objects in the library. This can be done using the GRTOBJAUT command of the form:

```
GRTOBJAUT OBJ(library/*ALL) OBJTYPE(*ALL) USER(proxyuser)
    AUT(*EXCLUDE)
```

Portal approach: An alternative to restricting access to all objects in the library is to use a “portal” approach. With this method, you restrict access at the library level only and define an alternative library for users to access instead of the main data files. For example, the main data library might be called PRODUCTION. Using the portal method, you would exclude the proxy user profile from the PRODUCTION library and instead grant access to another library called QRYPROD. Within QRYPROD, a set of logical files or SQL VIEWS would be defined that the proxy users could access. This approach has several advantages:

- You can avoid putting private authority on a potentially large number of objects.[1]
- The view of the data can be greatly simplified for end-users – unnecessary files are not visible, joins can be predefined, the number of fields available can be greatly reduced, and understandable file or field names can be used.
- Data level security can be defined through select/omit criteria on a logical file or a WHERE clause in a view.
- Common functions (like calculating discounts) can be performed in a view, thus eliminating work at query definition time.

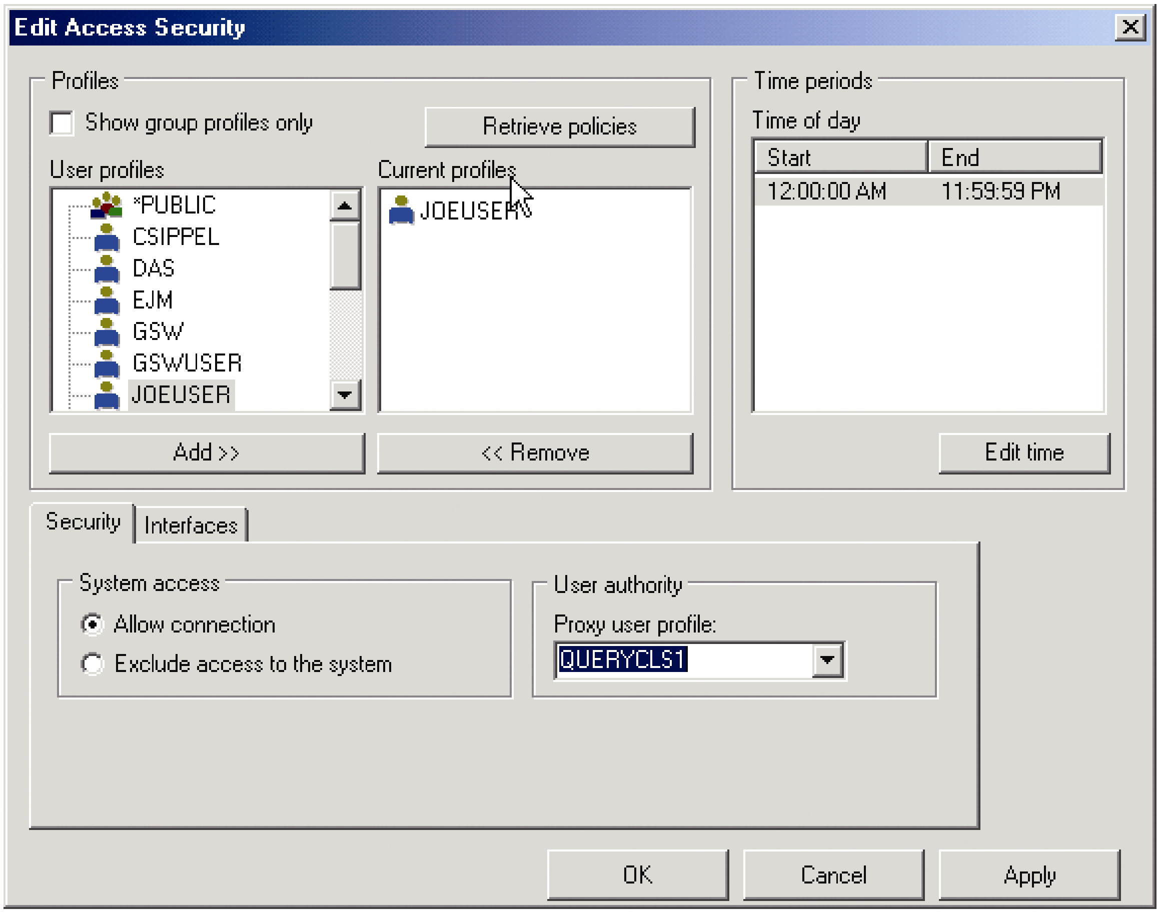
To summarize, these are the steps to create a class and assign authority:

[1] If you define a private authority of *EXCLUDE while *PUBLIC has more authority (like *CHANGE), there is a small performance penalty when the object is accessed as the system makes sure the current user is not excluded from the object.

1. Create one user profile for each class identified in step 2.
2. Restrict access to all but the necessary libraries by excluding the proxy user profile from those objects.
3. Use one of the following approaches to restrict access to objects in those libraries:
   - Exclude method
     i) Use the GRTOBJAUT command to exclude all objects in the library from the proxy user profile.
     ii) Use the GRTOBJAUT command to grant authority to the objects which should be made available to the proxy profile
   - Portal method
     i) Create a library which the proxy user profile has authority to access
     ii) Create logical files or views that reference the needed physical files
     iii) Assign appropriate authority to the views or logical files appropriate to the proxy profile's class

3.5.5.4 Step Four: Associating users with proxy user profile

The next step is to associate a user or group profile to a proxy. Once the association has been done, a user assigned to that proxy will use that profile's system authority rather than their own. The following dialog shows how you make the association in insure/Security between the user profile and the proxy user:



□ Show group profiles only

User profiles: *PUBLIC, CSIPPEL, DAS, EJM, GSW, GSWUSER, JOEUSER

Current profiles: JOEUSER

Time of day

Start	End
12:00:00 AM	11:59:59 PM

System access

● Allow connection

○ Exclude access to the system

User authority

Proxy user profile: QUERYCLS1

In this example, the user profile JOEUSER has been assigned to a proxy user of QUERYCLS1. The next time JOEUSER connects to the System i, he will only see objects the QUERYCLS1 user has authority to access.

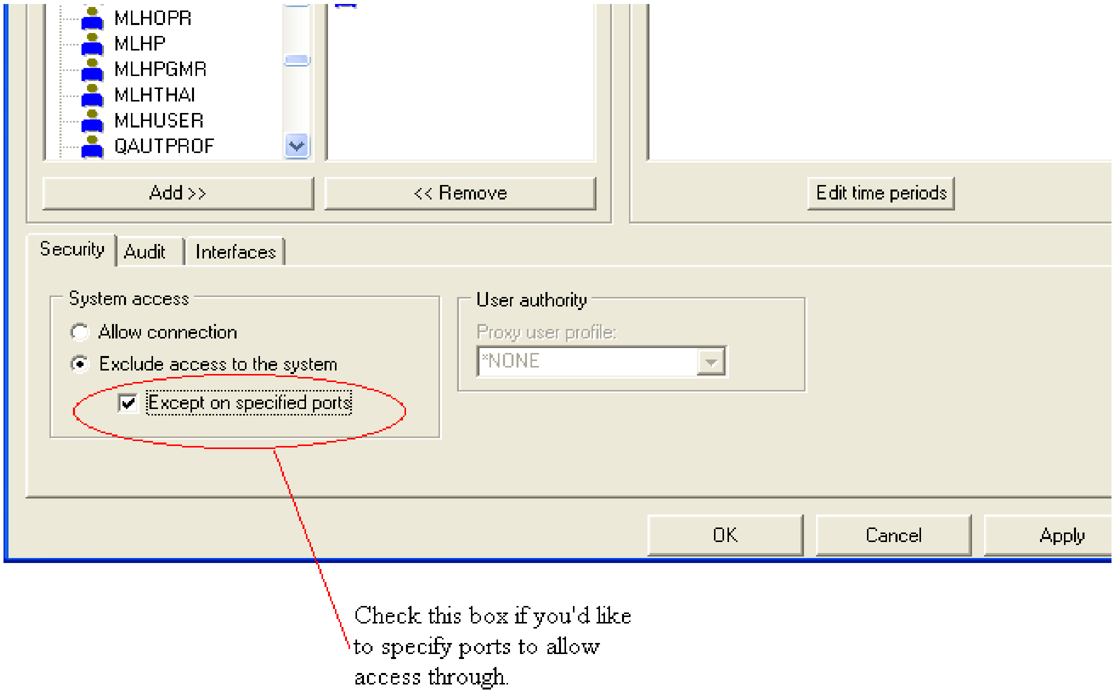
If you want the proxy user to only be in effect for certain hours, simply adjust the time periods specified in the upper right of the dialog.

3.5.5.5 Step Five: Testing authority settings

The final step is to verify the authority has been properly defined, assigned to the proxy user, and works for the users associated with that class. The easiest way to test the assigned authority is to use a desktop query tool (like Microsoft Access) and determine if the appropriate objects can be found and accessed at the assigned levels.

3.5.5.6 Step Six: Securing connections by port (Optional)

If needed, you can restrict users to specific port to control access to the system. This feature is required for regulations such as PCI (Payment Card Industry) Compliance. In the case of PCI compliance, insure/SECURITY can be configured to allow access to the interface through a particular port, in this case a secure port. See the following screens that show configuration steps:



MLHOPR
MLHP
MLHPGMR
MLHTHAI
MLHUSER
QAUTPROF

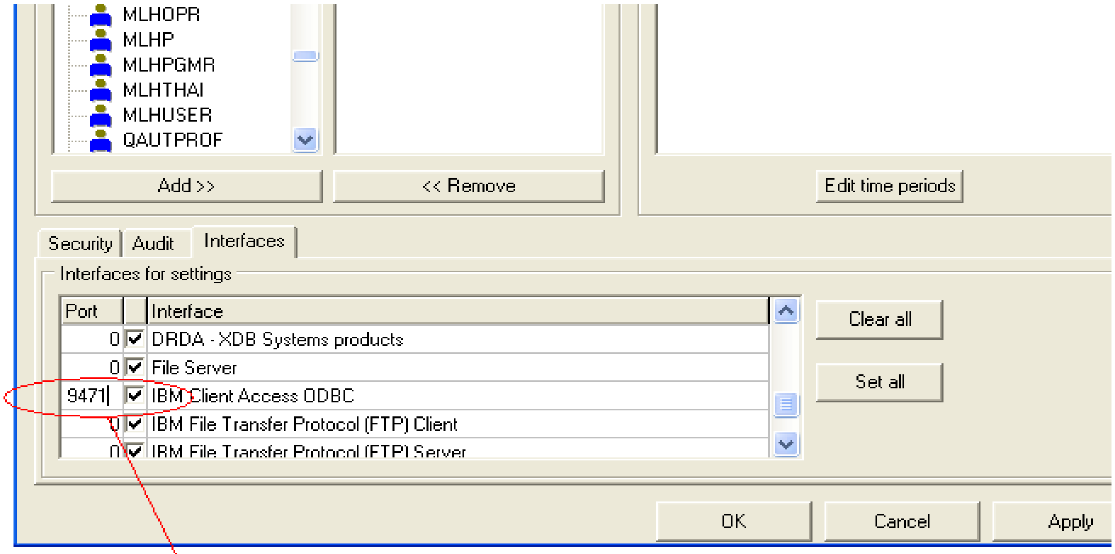
System access

○ Allow connection
● Exclude access to the system
☑ Except on specified ports

User authority

Proxy user profile: *NONE

Check this box if you'd like to specify ports to allow access through.



Port		Interface
0	☑	DRDA - XDB Systems products
0	☑	File Server
9471	☑	IBM Client Access ODBC
0	☑	IBM File Transfer Protocol (FTP) Client
0	☑	IBM File Transfer Protocol (FTP) Server

Specify the particular port to allow access on. 0 is a special value, signifying that no port checking is in place on the interface.

By specifying a port number, users that are assigned the policy will be required to use that port to connect with that specific interface.

3.6 Activity reports

The activity reports that come with insure/SECURITY provide an easy way to extract information about the use of your system and database. Security reports allow you to answer the following questions:

- What are my current security policies?
- Who has attempted to connect to my system and been rejected because of security policies?
- When have rejected connections been attempted?
- Operation details by user (i.e. statements executed)?

To access the built-in reports, choose ***Tracking Reports*** from the insure/Security section of the Welcome dialog. Alternatively, you can choose the *Tools* | *Administration tasks* | *Usage reports* menu option.

4 Support

Centerfield Technology Inc. is committed to providing our customers with support as problems or questions arise. Support includes minor enhancement releases and upgrades which might be necessary as OS/400 and i5/OS releases become available.

4.1 Contact Information

Internet
Web support pages are maintained at www.centerfieldtechnology.com. Updated documentation, how-to's, FAQs, and a list of known problems will become available as needed.

E-mail
To contact technical support by email, send email requests to support@centerfieldtechnology.com.

Fax
To contact technical support by fax, send your request to **(888) 908-3073**.

Phone
You can call 507-287-8119 for technical support. Telephone support is available from 8:00 AM to 5:00 PM CST.

4.2 Additional Information

- Visit Centerfield Technology's HomeRun web site www.centerfieldtechnology.com to find the latest software patches for HomeRun. Follow the support link from the home page.
  PTFs listed in this document are current as of the date this document was released. There are times where IBM releases PTFs applicable to the functions our software uses after the document release date. Accordingly, we strongly recommend you visit our website periodically and verify that all listed PTFs are applied on your system.

Appendixes

4.3 System-wide installation impacts

HomeRun changes some settings on your IBM System i™ system at installation time that can affect your current system configuration and activity. The HomeRun modifications that may have system-wide impacts are limited to:

- Changing system values
- Ending currently active Centerfield servers, jobs, and monitors
- Configuring network attributes
- Modifying subsystem configurations
- Adding exit points

This section also details information on HomeRun's use of the following IBM System i™ features:

- TCP/IP usage
- Job scheduler usage
- Authorities on installed libraries
- CL Commands and APIs used
- Programs which adopt authority

4.3.1 System Values

The HomeRun toolset helps you handle issues like performance, security, and usage tracking which requires special system privileges. Because the product installs software that performs privileged operations, it requires certain system values that control authority and application privileges to be set to certain values. The following table shows the list of system values and a description of the HomeRun installation requirements.

QALWUSRDMN	This system value is checked at installation to ensure that the value is either *ALL or that it contains the *{PROGRAM LIBRARY}* library as one of the libraries that allows user domain objects. HomeRun requires this setting because it stores and directly accesses user space objects in the *{PROGRAM LIBRARY}* library. If the system value is not set properly for HomeRun, the installation support modifies the setting to include the *{PROGRAM LIBRARY}* library.

QALWOBJRST	This system value is checked at installation to ensure that the value is either *ALL, or that it at least allows system state objects (*ALWSYSSTT) and objects that adopt owner authority (*ALWPGMADP) to be restored. HomeRun requires this setting because it restores objects with these attributes into the *{PROGRAM LIBRARY}* and *{DATA LIBRARY}* library. If the system value is not set properly for HomeRun, the installation will fail.
QUSEADPAUT	This system value is checked at installation time to ensure that the system does not restrict the list of user profiles that are allowed to modify programs to use adopted authority. HomeRun requires this because depending on the system program temporary fix (PTF) level, this system value can also control users that are allowed to run programs that adopt *OWNER authority. If the system value is not set properly for HomeRun, the installation will fail.

HomeRun references several other system values at run time to determine information about your system configuration. The following are other system values that are referenced at run time:

- QACGLVL
- QDATE
- QDAY
- QYEAR
- QMONTH
- QSRLNBR

4.3.2 Servers, Jobs, and Monitors

You need to ensure that the objects that will be replaced by the HomeRun installation and the resources that are being modified by the installation are not used or locked by other jobs on the system prior to starting the product installation. HomeRun creates and replaces objects in the *{PROGRAM LIBRARY}* and *{DATA LIBRARY}* libraries. You should ensure that there are no objects locked in these libraries prior to installing the software. However, even if you do not check these libraries before you start the installation, the HomeRun installation checks that some of the common jobs that can hold critical locks or cause other problems are always ended before letting the installation continue.

4.3.3 Database Monitor and HomeRun

Several features within HomeRun build on top of the IBM System i™ database monitor support (STRDBMON command). The HomeRun installation ends the database monitor if it is actively collecting system-wide activity. It ends the monitor to help ensure that there are no locks on files within the *{DATA LIBRARY}* library.

4.3.4 Subsystems and Work Management

The HomeRun server software handles:

- Managing and processing incoming requests from the administrative console software
- Enforcing user access policies configured by the administrative console software
- Scheduling of any activity requested by the administrative console
- Maintaining data collection information

The HomeRun server uses its own work management configuration to control how HomeRun client jobs are started. At installation time or using the HomeRun CFGSVR IBM System i™ command, you define the subsystem that HomeRun activity should run in. By default the Centerfield server will run in its own subsystem defined by the XCSBS80 subsystem description in the program library. It is highly encouraged that you use this subsystem to isolate the Centerfield server from other system work. At installation time, an autostart job entry is put into the QUSRWRK subsystem. This autostart entry will automatically start the Centerfield subsystem and server so you do not normally have to start the server manually after an IPL. **If you use the default subsystem (XCSBS80) you can skip the rest of this section.**

The job queue is used as the starting point for starting new HomeRun client jobs. The job queue entry that is added is for the XCTCP job queue that is found in the *{PROGRAM LIBRARY}* library. If you are changing an existing configuration or installing over a previous version of a HomeRun server, the existing job queue entry is first removed and then the new entry is added to the specified subsystem. The command that HomeRun uses to add the job queue entry is similar to the following. The default subsystem is XCSBS80, and the sequence number begins at 50. If 50 is not available, the number is automatically incremented by the HomeRun configuration support until an unused sequence number is found.

> ADDJOBQE SBSD(**subsystem description**) JOBQ(*{PROGRAM LIBRARY}*/XCTCP) MAXACT(*NOMAX) SEQNBR(50)

A secondary job queue entry, XCAUTODBA, is used to schedule background work for HomeRun's AutoDBA feature. By default it is placed at SEQNBR(60).

The routing entry that HomeRun installs controls the routing of the HomeRun client jobs. It must be added to the same subsystem that has the XCTCP job queue entry. The routing entry uses the XCTCP routing data for comparison and calls the QCMD program. The XCTCP job class is specified as the job class for the routing entry. The command that HomeRun uses to add the routing entry is similar to the following. The default subsystem is XCSBS80, and the sequence number begins at 170. If 170 is not available, the number is automatically incremented by the HomeRun configuration support until an unused sequence number is found. The second routing entry is added for work that should be run at a lower priority level and therefore uses a different class than the rest of the server jobs.

> ADDRTGE SBSD(**subsystem description**) SEQNBR(170) CMPVAL(***varies by release***) PGM(QSYS/QCMD) CLS(*{PROGRAM LIBRARY}*/XCTCP)
>
> ADDRTGE SBSD(**subsystem description**) SEQNBR(180) CMPVAL(***varies by release***) PGM(QSYS/QCMD) CLS(*{PROGRAM LIBRARY}*/XCTCPBJ)

If you are changing an existing configuration or installing over a previous version of HomeRun, the existing routing entry is not removed. It should not cause any harm to your existing application environment to leave it installed in a subsystem that is no longer used by HomeRun. If you want to remove it, you need to remove it using standard work management support available on the IBM System i™. You can use a command similar to following. Before issuing the command, you need to make sure that the sequence number used on the remove routing entry command is the sequence number for the HomeRun routing entry. You can check the routing entries by using the WRKSBSD command and displaying the routing entries for the subsystem that you want to change. For more information about performing work management activities, see IBM System i™ work management documentation.

> RMVRTGE SBSD(**subsystem description**) SEQNBR(170)
>
> RMVRTGE SBSD(**subsystem description**) SEQNBR(180)

The HomeRun installation creates objects within the *{PROGRAM LIBRARY}* installation library to support these and other work management changes on your system. The following are objects that are created by the HomeRun installation support that can be used for work management control.

Object Name	**Object Type**	**Description**
XCTCP	Job description	Job description used to set job properties for HomeRun administration client jobs
XCTCP	Job class	Job class used to set job properties for HomeRun administration client jobs
XCTCPBJ	Job class	Jobs class used to run jobs that may run long- running operations and should execute at a lower priority than other work.
XCTCP	Job queue	Job queue used to start HomeRun administration client jobs

XCHLP	Job description	Used by the XCHELPER background job
XCIAFINDER	Job description	Used by the XCIAFINDER background job (V5R4 and higher)
XCPC	Job description	Used by the XCPC background job
XCSTRSVR	Job description	Used by the autostart job entry
XCSCRUBBER	Job description	Job description used to set job properties for the HomeRun Collection Scrubber job
{PROGRAM LIBRARY}	Output queue	Output queue used for upcoming product enhancements

You can modify the objects used for controlling work management, but your changes will not be preserved when you upgrade to the next version of HomeRun.

4.3.5 Exit points

The following exit point is added during HomeRun product installation:

Exit Point	HomeRun Program Name
QIBM_QWT_JOBNOTIFY	XCDTAQ

This exit point is used by the Usage Tracker for insure/INDEX and insure/ANALYSIS. At installation time, no jobs are ended and no subsystems are ended. However, if you want to have the Usage Tracker monitor for specific new jobs that start (that is, if you choose the ***Filtered jobs*** Profile type), you will need to end and restart any subsystems in which those jobs might start before your monitor will take affect. This only needs to be done once after the installation of HomeRun.

Three other exit points are optionally added at installation time. A prompt will appear that asks if these exit points should be added so that additional types of data collection can be done by HomeRun. By default they are installed, but they can be bypassed if so desired. The three exit points are:

Exit Point	HomeRun Program Name
QIBM_QSQ_CLI_CONNECT	XCCLIINIT
QIBM_QZDA_INIT	XCODBCINIT
DDM exit point (DSPNETA)	XCDDM

4.3.6 TCP/IP Usage

The HomeRun server uses TCP/IP to communicate with the HomeRun clients. The server and client applications communicate almost exclusively using a TCP application. The application is written to use proprietary application data flows to give fast and efficient performance.

The IBM System i™ and personal computer sockets applications communicate with each other through TCP ports. Ports are used by the TCP protocol to identify a unique origin or destination of communication with a TCP application. TCP ports can be any numeric value from 1 to 65535. TCP applications that are commonly used, like ftp and telnet, use pre-assigned port numbers. Pre-assigned port numbers are called well-known TCP ports. The well-known TCP port numbers range between 1 and 1023 and should not be used when you configure HomeRun. If you specify one of these ports, it can affect the operation of the application that normally uses the well-known port. When you install HomeRun, a port and associated application service is configured. The service name for HomeRun is CENTERFIELD_SERVER_80 and the port by default is **{default port}**. If the **{default port}** port is already used by another application, the port number is incremented until a free port is found. The command that HomeRun uses to add the service and port configuration is similar to the following. You can check the port and service configuration after you install by using the WRKSRVTBLE command and locating the CENTERFIELD_SERVER_80 in the service list.

ADDSRVTBLE SERVICE(CENTERFIELD_SERVER_80) PORT(**{default port}**)
PROTOCOL('tcp') TEXT('Centerfield Technology Server')

In addition to the proprietary communications method, HomeRun uses an ODBC connection to handle report requests made by the administration client. HomeRun requires an iSeries Navigator ODBC connection. When the HomeRun client is installed, the auto-configuration support will attempt to configure an ODBC data source if the supported ODBC driver can be detected.

4.3.7 Job Scheduler Usage

The IBM System i™ built-in job scheduler is used by several components of HomeRun. Events that can cause jobs to be placed on the job scheduler are:

- Scheduling a Database Profiler data collection
- Scheduling Index Create requests using insure/INDEX or Visual SQL Explain
- Cleaning database collections using the Collection Scrubber
- Autonomic Database Assistant (AutoDBA) analysis and actions

4.3.8 Default public authority of libraries

The *{PROGRAM LIBRARY}* and *{DATA LIBRARY}* libraries are created with the default create authority set to *CHANGE.

4.3.9 Command Language (CL) Commands Used by HomeRun

HomeRun is continually being enhanced and modified, so the list of CL commands used by HomeRun also continues to change. The following is a list of CL commands that are used by HomeRun. If you have modified command defaults or installed an application that either replaces or changes any of the following commands, you should contact Centerfield Technology to discuss the impacts that the changes may have on HomeRun. **NOTE: This list may change without notice and is not guaranteed to be complete for the most current version of software.**

ADDPFTRG	CHKTAP	DLTJRN	IF	RTVMBRD
ADDMON*	CLOF	DLTJRNRCV	GOTO	RTVMSG
ADDEXITPGM	CPROBJ	DLTOVR	MONMSG	RTVNETA
ADDJOBQE	CPYF	DLTPGM	MOVOBJ	RTVOBJD
ADDJOBSCDE	CPYSPLF	DLTSP*	OPNDBF	RTVSYSVAL
ADDMSGD	CRTDTAARA	DLTSPLF	OVRDBF	SAVOBJ
ADDRTGE	CRTDUPOBJ	DLTUSRSPC	OVRPRTF	SBMJOB
ADDSRVTBLE	CRTJRN	DO	PASSWORD*	SNDPGMMSG
ALCOBJ	CRTJRNRCV	DSPDBR	PGM	SNDRCVF
CALL	CRTLF	DSPFD	PRTSQLINF	STRDBG
CALLPRC	CRTLIB	DSPJOB	RETURN	STRDBMON
CFGSVR*	CRTOUTQ	DSPJOBLOG	RCVF	STRHOSTSVR
CHGACGCDE	CRTPF	ENDDBG	RCVMSG	STRJRNPF
CHGJOB	CRTSAVF	ENDDBMON	RMVJOBQE	STRSVR*
CHGLIBL	CRTSP*	ENDDO	RMVJOBSCDE	STRTCPSVR
CHGNETA	CRTSRCPF	ENDHOSTSVR	RMVMSGD	
CHGOBJOWN	CRTUSRPRF	ENDJOB	RMVPFTRG	
CHGQRYA	DCL	ENDJRNPF	RMVMON*	
CHGSYSLIBL	DCLF	ENDPGM	RNMOBJ	
CHGSYSVAL	DLCOBJ	ENDPJ	RSTOBJ	
CHGVAR	DLTDTAARA	ENDSVR*	RTVDTAARA	
CHKOBJ	DLTF	ENDTCPSVR	RTVJOBA	

* Indicates Centerfield Technology command

4.3.10 OS/400 System APIs Used by HomeRun

HomeRun is continually being enhanced and modified so the list of system application interfaces (APIs) used by HomeRun also continues to change. The following is a list of system APIs and header files that contain API interfaces that are used by HomeRun. If you have installed programs that either replace or change any of the following APIs, you should contact Centerfield Technology to discuss the impacts that the changes may have on HomeRun.

NOTE: This list may change without notice and is not guaranteed to be complete for the most current version of software. This list may not include every API that is used if the API name does not match the included source member (i.e. the name of the API is in the source file and used by Centerfield but not explicitly listed here).

CEELOCT	QUSDLTUS	QWTSETP
QCMDEXC	QUSGEN	QWCRSVAL
QDBRTVFD	QUSLJOB	QUSRTVEI
QDBLDBR	QUSLMBR	QUSMBRD
QLICHGLL	QUSLOBJ	QUSRTVFD
QMHRTVM	QUSPTRUS	QUSCHGPA
QMHSNDPM	QUSRJOBI	QDBBRCDS
QSYGETPH	QUSRMBRD	QTNADDCR
QSYRLSPH	QUSRTVUS	QUSMIAPI
QUSCHGUS	QWCRJBST	QDMLOPNF
QTNRMVCR	QWCRSSTS	QMHRSNEM
QUSEC	QWCLOBJL	QPMWKCOL
QJOURNAL	QMHRCVPM	QWDLSJBQ
QTOCNETSTS	QPMLPFRD	QWCLSCDE
QWCCHGTN	QWDLSBSE	
QPMLPMGT	QUSCUSAT	

4.3.11 Programs Adopting *OWNER Authority

The following programs adopt *OWNER authority. All HomeRun programs are compiled to use adopted authority.

Program Name	**Description**
XCACTODBC	Retrieve actively connected user access jobs
XCADDJE	Add job notify exit point
XCADDJOBCFG	Add job configuration command processor
XCADDUAE	Add the remote monitoring support
XCADVIX	Advise indexes
XCAUDOPR	Audit a client request

XCJBCFGPOP	Change job history prompt override processor
XCCFGMONHS	Configure monitor history auditing
XCCFGMONHP	Configure monitor history prompt override program
XCCHGODBCA	Change run time attributes of a user access job
XCCHKCMTCTL	Check commitment control driver program
XCCMD	Internally used by client software to run a Command Language (CL) command
XCCMT (service program)	Commitment control API interfaces
XCDSPLICINF	Display system license information
XCENDHLPCP	End helper command processing program
XCENDSCRBR	End scrubber job command processor
XCHELPER	Background job reading job notify data queue
XCHLP	Main helper job program
XCIAFINDER	Read journal to retrieve recommended indexes
XCIP	Capture IP address for job
XCJOBUTIL (service program)	Job APIs that require job control authority
XCJDELOG	Look at JDE logs
XCJDEINI	Edit JDE.INI file
XCJOBPRF	Start a database monitor collection for a job
XCLOCKCF	Lock APIs
XCLSTJOB	List jobs APIs
XCMONHST	Monitor job history to audit table
XCMTXWAIT	Mutex APIs
XCPASS	License entry and verification
XCPERFCOLL	Performance collector APIs
XCPMSUB	Dynamically substitute parameters for a parameterized SQL SELECT statement
XCPRFCRT	Create a database collection profile
XCPRFDTL	Delete a database collection profile
XCPRFEND	End a Database Profiler collection
XCPRFENDAP	End all active Database Profiler collections
XCPRFRNM	Rename a database collection profile
XCPRFSTR	Start a Database Profiler collection
XCOWCFG	Configure server for OneWorld environment
XCREPBLD	Build Database Profiler repository
XCREPDLT	Delete repository

XCRTVCFGVAL	Retrieve configuration value
XCRMVJBCFG	Remove job configuration
XCRMVUAE	Remove the remote monitoring support
XCSCDTRG	Schedule requested work (used as trigger program for XCSCD file)
XCSCRUB	Remove unneeded collection data
XCSEMWAIT	Semaphore APIs
XCSIGNON	Check password at sign on time and swap user profile to signed in user
XCSPDSPCMD	Internally used by client software to run an OS/400 command and return spool file output
XCSQLEXEC	Internally used by client software to run non-SELECT SQL statement with commitment control
XCSQLEXEC0	Internally used by client software to run non-SELECT SQL statement without commitment control
XCSQLSEL	Internally used by client software to run SQL SELECT statement with system naming
XCSQLSEL2	Internally used by client software to run SQL SELECT statement with SQL naming
XCSTRHLPCP	Start helper command processing program
XCSTRSVR	Start the Centerfield server
XCSYSUTIL (service program)	System information APIs
XCTRIGGER	Trigger program for configuration files
XCTRIGGERM	Trigger program for job monitor configuration files
XCDBGUTIL (service program)	Debug utilities

4.4 Enabling remote monitoring

To enable the remote monitoring functionality of some of the HomeRun tools, the ADDMON command must be run on your IBM System i™ server. You should enable remote monitoring if you want to:

1. use insure/MONITOR to monitor and track usage by users of remote interfaces, such as ODBC or FTP
2. use insure/SECURITY to prevent access by unauthorized users of remote interfaces.

4.4.1 Effects of exit point registration on your environment

The ADDMON command can be found in the HomeRun server installation library. The command allows you to specify the interfaces that you want to monitor and control. See your documentation for insure/MONITOR or insure/SECURITY for a list of supported interfaces.

When the ADDMON command is run, some control information is created within the installation library, and exit programs for the requested interfaces are registered on the system. For all registration facility exit points used by HomeRun, any currently configured exit programs will be replaced with HomeRun's exit programs. This is done because the operating system only honors one exit program per exit point for several of the available exit points. If an exit program was previously configured, the installation support saves the reference to the exit program in an internal area. When the HomeRun exit program is called, HomeRun's security, resource, and auditing policies will be enforced. If a user exit program was configured before HomeRun was installed, and HomeRun's policies did not reject the connection, the previously defined user exit program will be called *in addition* to the HomeRun programs, and its output will be returned as if it had been called directly.

If the DDM access exit point had been configured to a value other than the default, it is handled in a similar manner. HomeRun's exit program is called and the configured HomeRun policies are enforced. If a non-default value was detected for the DDM access exit point at HomeRun installation time, and HomeRun did not reject the connection, either the user defined exit program will be called or the previously defined action will be taken. The output returned from the exit program will be functionally consistent to what would have been returned had HomeRun not been installed.

In some cases you may want HomeRun's exit program called *after* your existing exit program. You may or may not be able to make this work depending on the design of your existing exit program and the method that you use to install your program.

- If your exit program is registered by installing a third party application, you need to ensure that the third party application has a method for remembering and calling exit programs that exist on an exit point prior to their installation. You should also ensure that when the third party application is removed that the application restores the exit point to its original state.

- If your exit program is a custom program designed by your own staff and installed directly using operating system commands, you need to ensure that you have designed it to have a method for knowing about and calling HomeRun's exit program after it has done its processing.   In both cases, the HomeRun exit program must be passed the same parameters that it would have been passed if the operating system had called the program directly. If all of your exit programs meet these criteria, you can achieve any exit program call order that you would like. The call order will be a Last In First Out (LIFO) ordering scheme. The last exit program installed will be the first exit program called when the exit point is encountered. To force HomeRun's exit program to be called last you need to:

  1. De-register all of the user exit programs that you currently have configured for each exit point that the HomeRun server uses. See the next section to determine the specific exit points used by HomeRun. If you do not know whether you use exit programs today, you can use the WRKREGINF and DSPNETA commands to display the currently registered exit programs for the various exit points.
  2. Run the ADDMON command to register the HomeRun exit programs and enable the remote monitoring support.
  3. Re-register each of your user exit programs. If you have multiple exit programs to install on the same exit point, you should record the order that you use to install them to ensure that if you need to remove them that you use exactly the reverse order.

4.4.2 Exit Program Registration Details

HomeRun registers exit programs using the IBM System i™ registration facility support for several products and features. You control the exit programs that are installed by specifying parameters on the ADDMON command.

The exit programs that are registered by HomeRun enforce the HomeRun policies. The installation of an exit program can affect your current system activity. Most of the OS/400 products and features that allow exit programs to be registered do not fully reinitialize when an exit program is installed. The issues surrounding how exit programs are added, resolved, and called can cause run time failures if an exit program is replaced after a job has initialized the calling information for an exit program. ***For this reason, any system activity for the products and features that have exit points that are modified by the HomeRun support should be ended prior to running the command, and not resumed until the command completes.***   If any job using these products or features remains active or becomes active through the course of the installation, unexpected run time errors may occur the next time that the job calls the registered exit program. ***Note:*** IBM documentation states that if the exit program for File Server (QIBM_QPWFS_FILE_SERV) is changed, the QSERVER subsystem must be ended and restarted for the changed to take effect.

To manually end OS/400 products and features that have exit points that will be modified by HomeRun, you can use the following commands and techniques:

- To end FTP:
  QSYS/ENDTCPSVR SERVER(*FTP)

- To end IBM ODBC, first issue this command:
  QSYS/ENDHOSTSVR SERVER(*DATABASE)

  Then do one or both of the following:
  If your ODBC users connect over APPN use this command:
  QSYS/ENDPJ SBS(QSERVER) PGM(QIWS/QZDAINIT)

  If your ODBC users connect over TCP/IP use these commands:
  QSYS/ENDPJ SBS(QSERVER) PGM(QIWS/QZDASOINIT)

  QSYS/ENDPJ SBS(QSERVER) PGM(QIWS/QZDASSINIT)

To restart the servers after the exit points are installed, issue the following commands:

- STRHOSTSVR SERVER(*DATABASE)

- STRTCPSVR SERVER(*FTP)

Exit points that the ADDMON command modifies are specified in the following table.

Exit Point	ADDMON option specified
QIBM_QZSC_SM	*CSCM
QIBM_QZSC_NLS	*CSCONV
QIBM_QZSC_LM	*CSLM
QIBM_QZHQ_DATA_QUEUE	*DTAQ
QIBM_QNPS_ENTRY	*NETPRT
QIBM_QVP_PRINTERS	*VRTPRT
QIBM_QHQ_DTAQ	*DTAQORIG
QIBM_QTF_TRANSFER	*FILETRANS
QIBM_QLZP_LICENSE	*LMORIG
QIBM_QMF_MESSAGE	*MSGORIG

QIBM_QRQ_SQL	*RMTSQL
QIBM_QZRC_RMT	*RMTCALL
QIBM_QZRC_RMT	*RMTCMD
QIBM_QPWFS_FILE_SERV	*FILSRV
QIBM_QTMX_SVR_LOGON	*REXEC
QIBM_QZSO_SIGNONSRV	*SOCKSRV
QIBM_QTG_DEVINIT	*TELNET
QIBM_QZDA_SQL1, QIBM_QZDA_SQL2	*IBMODBC
QIBM_QTMF_SERVER_REQ, QIBM_QTMF_CLIENT_REQ	*IBMFTP
QIBM_QZDA_INIT	*ODBCINIT
QIBM_QSQ_CLI_CONNECT	*CLIINIT
QIBM_QZDA_INIT QIBM_QSQ_CLI_CONNECT DDMACC in CHGNETA command	*INSURE_IA

Some data access interfaces like DRDA based ODBC do not support the registration facility. To support DRDA based data access, HomeRun registers the XCDDM exit program under the network attributes DDM access exit point. On V4R1 and above, the IBM System i™ DRDA support calls the DDM access exit program when a connection is established. Prior to V4R1, the exit program is not honored for DRDA. If the DDM access setting is a value other than *OBJAUT prior to installing the E-Connect Server, the installation support will honor the previously configured value in addition to the E-Connect Server's configured policies.

Interfaces that are controlled via the DDM access exit point are shown in the following table. To specify these interfaces on the ADDMON command, use the *IBMDDM parameter.

Distributed Data Management
DRDA - FileTek products
DRDA - Grandview DB/DC Systems products
DRDA - IBM DB2 Connect (formally DDCS)
DRDA - IBM DB2 for VSE and VM
DRDA - IBM DB2 UDB for IBM System i™
DRDA - IBM DB2 UDB for OS/390
DRDA - Informix Software products
DRDA - Oracle Corporation products
DRDA - StarQuest products
DRDA - Wall Data Rumba for Database Access
DRDA - XDB Systems products
DRDA - Derby Network Client
DRDA - Java Client (JCC)
DRDA – DataDirect Technologies
DRDA - SeeBeyond ICAN (Sun JCAPS)

If manual steps for recycling servers in question are a hassle, there is an ‘automated’ alternative. At the end of “HomeRun Installation Guide” document there is a section titled “Recycle host servers sample CLLE script” which outlines steps to create a RECYCLESVR program which can then be invoked to recycle the host servers upon running of RMVMON command. Sample code refers to default installation settings.

4.5 Removing remote monitoring support

The HomeRun remote monitoring support can be disabled by running the RMVMON command that can be found in the HomeRun installation library. The command allows you to disable the support for interfaces that you specify. The default value is to remove all interfaces.

If you registered additional exit programs on the same exit points used by HomeRun after the HomeRun server was installed, you need to ensure that you de-register the additional exit programs *before* disabling the HomeRun remote monitoring support. If any registration facility user exit programs were replaced when the HomeRun remote monitoring support was added, they will be restored to their original values. Likewise if the DDM access exit point had been configured to a value other than the default prior to the registration of HomeRun remote monitoring support, it will be restored to its previous value.