Enterprise Application Remote Access

Guide

UNISYS

Release 3.3

March 2003 78616562-003

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product or related information described herein is only furnished pursuant and subject to the terms and conditions of a duly executed agreement to purchase or lease equipment or to license software. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, special, or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Notice to Government End Users: The software and accompanying documentation are delivered and licensed as "commercial computer software" and "commercial computer software documentation" as those terms are used in 48 C.F.R. § 12.212 and 48 C.F.R. § 227.7202-1 through 227.7202-4, as applicable. The Government shall receive only those rights provided in the standard commercial software license, or where applicable, the restricted and limited rights provisions of the contract FAR or DFARS (or equivalent agency) clause.

Correspondence regarding this publication should be forwarded to Unisys Corporation by addressing remarks to Product Information, Australian Centre for Unisys Software, Locked Bag 3111, Rhodes, NSW 2138, Australia. Comments about documentation can also be sent through e-mail to **eae.pi@unisys.com**.

Contents

Section 1
Introduction

This section describes the purpose of this document as well as the intended audience. It also gives an overview of the Remote Access server.

The following topics are covered in this section:

- "About this Guide"
- "Introduction to the Enterprise Application Remote Access Server"

About this Guide

Purpose

The purpose of this document is to provide information on the installation, configuration, and administration of the Remote Access server with Enterprise Application Developer, Developer Test, and Runtime on the following supported platforms:

- Windows operating systems
- UNIX operating systems
- ClearPath MCP
- ClearPath OS 2200

Audience

This document is written for users of Component Enabler who are responsible for installing and configuring the server.

The audience is presumed to have a working knowledge of Runtime on their specific platform as well as either Component Enabler.

Introduction to the Enterprise Application Remote Access Server

The Remote Access servers can be used by Component Enabler to access Systems.

The server-side deployment consists of a number of platform-specific Remote Access servers that handle communications between clients and applications. Clients and servers communicate, using RATL protocol running on TCP/IP. This protocol is a simple wrapping of a standardized and extended set of NOF messages that provide all the functionality needed to support GUI forms.

Clients establish normal sessions with applications through the Remote Access servers using a TCP/IP connection. These servers perform a number of checks, such as authenticating users and verifying the application exists, before granting the client access. The servers then receive transaction requests from client programs over a TCP/IP connection and pass them to the application. Transaction responses from applications are passed back to the clients over the same TCP/IP connection. Remote Access servers can also pass back asynchronous messages, such as messages from Reports, to a client application.

Remote Access servers can also use the Microsoft Message Queue (MSMQ) server for connections instead of TCP/IP connections.

The Remote Access servers are designed to offer connectivity, throughput, and transit times comparable to terminal access to the applications on the same host Systems. Each client application runs independently of all other clients, with its own connection back to the application server. There are no points of single threading or possible congestion in the transaction path to add overheads or reduce performance.

Caution!

Risks with Connection Pooling and Timeout:

If connection pooling is used for the connection to the Remote Access Server and a timeout occurs during a transaction before a client receives a response to a request, when the connection is returned to the pool it is possible for another client to pick up that connection, connect to the same server and receive the response intended for the previous client.

For information about Connection Pooling, refer to the Component Enabler Guide, section 13 Scalability for Component Enabler.

Section 2
Installing the Enterprise Application Remote Access Server

This section describes the software requirements and installation instructions for the Remote Access Server on each supported platform.

The following topics are covered in this section:

- "Installing the Remote Access Server on Enterprise Application Runtime Platforms"
- "Installing the Remote Access Server with Enterprise Application Developer"

Installing the Remote Access Server on Enterprise Application Runtime Platforms

Remote Access Server on Runtime for Windows Operating Systems

If MSMQ (Microsoft Message Queue Server) is installed on your Runtime for Windows host, then both Remote Access (MSMQ) Server and Remote Access (TCP/IP) Server are installed, otherwise only Remote Access (TCP/IP) Server is installed.

Note: *TCP/IPshouldbeconfiguredasdescribedin*UnisysEnterpriseApplicationGetting Started with Runtime for the Windows 2000 Operating Systems*.*

Remote Access Server on Runtime for the UNIX Operating System

If MQ (message queueing) software, such as Geneva Message Queueing Client for Java, is specified during the standard Runtime for UNIX installation procedure, then both Remote Access (MQ) Server and Remote Access (TCP/IP) Server are installed, otherwise only Remote Access (TCP/IP) Server is installed.

Remote Access Server on Runtime for ClearPath OS 2200

If TIP Session Control is installed on your Runtime for OS 2200 host, secure session access to Enterprise Application Environment Systems is automatically enforced.

To use Remote Access Server, you must have either CPCOMM or CMS 1100 (or both) installed.

Note: *CPCOMM is the default communications handler for Enterprise Application Runtime for OS 2200 release levels subsequent to Release 3R2.*

Remote Access Server on Runtime for ClearPath MCP

Codefile

The codefile is automatically installed to the CCF system pack.

CCF params file

- Obtain the sample CCF params file for the Remote Access server, named:

  `(<EAE usercode>)LINC17/SAMPLE/CCF/PARAMS/RATL ON <EAE dictionary>`

- Enter the Remote Access Entities required. For information on configuring entities see "Configuring the Remote Access Server on MCP" on page 3-13.
- Insert the definitions into the current copy of the CCF params file,

  for SSR 47.1, use **`SYSTEM/COMS/CCF/PARAMS ON <COMS CCF PACK>`**

  for SSR 48.1, use **`SYSTEM/CCF/PARAMS ON <CCF PACK>`**

  Note: *For A-Series Systems the CCF params file may not be present. Create this file using the sample provided,* ***`(<EAE usercode>)LINC17/SAMPLE/CCF/PARAMS ON <EAE dictionary pack>`***

COMS

Register the entities to COMS via the Utility window with the following file:

`(<EAE usercode>)LINC17/SAMPLE/COMS/CONFIG/RATL ON <EAE dictionary pack>`

Installing the Remote Access Server with Enterprise Application Developer

Software Requirements

To use the Remote Access Server, you must have:

- Enterprise Application Developer
- Component Enabler Client software

Installation

To install the Remote Access Server with Developer, select the Custom installation type from the Enterprise Application Developer Installation screen of the Setup program, and check the Developer Test Remote Server check box from the subsequent Select Component screen.

Each Remote Access Server is automatically configured to the following unique TCP/IP port numbers by default:

- Listener port – 4323

  This is the TCP/IP port to which the Component Enabler client is connected.

- Developer Test port – 4324

  This is the TCP/IP port used for interprocess communication between the Remote Access Server and Developer Test.

These port numbers can be configured to any other values, but you must ensure that the port numbers selected are not used by any other TCP/IP service.

Summary

After completing this section you can:

- Understand the Remote Access Server installation procedure on Runtime platforms.
- Follow the procedure to install the Remote Access Server with Developer.

Section 3
Configuring the Enterprise Application Remote Access Server

This section describes the configuration instructions for the Remote Access server on each supported platform.

The following topics are covered in this section:

- "Configuring the Remote Access Server on Windows Operating Systems"
- "Configuring the Remote Access Server on UNIX Operating System"
- "Configuring the Remote Access Server on MCP"
- "Configuring the Remote Access Server on OS 2200"
- "Configuring the Remote Access Server for Enterprise Application Developer"
- "Using the Remote Access Server with Graphical Interface Workbench"

Configuring the Remote Access Server on Windows Operating Systems

Before you can use the server, you need to configure it and create Views using the Remote Access page on the Environment Configuration property sheet in the Administration Client.

The Remote Access server configuration information regarding views, is stored in the Windows registry. Port number information is stored in the file NTLINC.CFG. You should not modify the registry or the port number file directly. Use the page in the Administration Client to safely add the data you need to configure the server.

Working With Views

Adding a View

1. Click the **Maintain Views** button, then click **Add**.
2. Type the View name, the name of the System, the path and directory of the System, and the Database ID used by the System.
3. Click the **Login** button to open the Login Screen Details dialog box. You can enter details for individual user ids, including:
    - Username label

- Username text
- Domain label
- Domain text
- Password label
- Password text

This information is displayed when you connect to the Component Enabler Viewer.

You can also access this information when you use the Component Enabler API and display them wherever you want.

4. Click the **Security** button to add security information using the Security Values dialog box.

   Server administrators have the following three options for constraining or permitting client connections to the System for a View:

   - Define a list of IP addresses, including IP ranges (123.123.12.*), that can be permitted connection or automatically forbidden connection. If you specify a list of permitted IP addresses, only users from that domain are allowed to access the View. Each entry should be separated by either a space or a semicolon (;).
   - Define a list of user ids that can be permitted connection or automatically forbidden connection. List the user ids in the format domain1\userid1; domain2\userid2; .... Where the user is defined on the local machine use .\userid. Use the radio button to specify the user ids that are permitted or forbidden to access the View. Each entry should be separated by either a space or a semicolon (;).
   - Select the **Anonymous** radio button to specify anonymous user support for the View. Specify an anonymous login that will initiate a connection to the System using the user id, ALPublic, and the password and domain name defined for ALPublic. Use the environment configuration dialog box in the Administration Client to enable anonymous logins. Set the environment variable `LINCII_ENABLE_ANONYMOUS` to the value "YES". This will enter the variable in your ntlinc.cfg file.

     Any Systems currently running must be restarted for the change to take effect.

5. Use the **Logging** radio buttons to select your logging options. You can set logging on or off, or you can set logging at the client's request.

Changing an Existing View

Click the **Maintain Views** button. The list window displays all the existing views, which you can remove or edit.

Note: *Adding and changing Views takes effect immediately. You do not need to restart the service.*

Invoking the Language Mapping Utility

1. Click the **Language Mappings** button.

   The Languages list box contains all the languages for all the defined Views. The ISO Languages list box contains a list of ISO-recognized languages. The ISO Countries list box contains a list of ISO-recognized countries.

2. Select a combination of ISO Country and ISO Language to uniquely identify the language dialect.
3. Select the Language to which you wish to map the selected dialect, then click the **Map Language** button.

Enabling Anonymous Login

1. Create the user ALPublic.
2. Click the **Public user details** button.
3. Use the Login details for user ALPublic dialog box to enter the password and domain name for use with this user id.
4. Define the ALPublic user id to the Windows User Manager.
5. In the Security Values dialog box, select the **Anonymous** radio button to specify anonymous user support for the View.
6. Specify an anonymous login that will initiate a connection to the System using the user id, ALPublic, and the password and domain name defined for ALPublic.
7. Use the Environment Configuration dialog box in the Administration Client to enable anonymous logins. Set the environment variable `LINCII_ENABLE_ANONYMOUS` to the value “YES”. This will enter the variable in your ntlinc.cfg file.

Enabling support for anonymous users allows multiple sessions to be started with the same System using one usercode. Login details do not need to be entered by the user.

Changing the WDP Repository Listener and Remote Access Port Number

Set the WDP listener port or Remote Access port to a value in the range 1001 — 65535. Do not enter the same number for both fields.

The Remote Access server contains a subset of WDP listener functionality which allows clients with no WDP process running to download items, such as images or list data files, from the Graphical Interface Workbench repository. Be sure to enter a port number that is not already assigned to any other process.

Message Queuing

MSMQ messaging allows multiple concurrent users of Component Enabler to access an Enterprise Application Environment System, by making more efficient use of system resources, and reducing the overheads associated with establishing and maintaining individual connections for each user session. This is an alternative to the existing method of connecting to the Runtime System using individual TCP/IP connections.

See your *Unisys Enterprise Application Component Enabler Developer's Guide* for further details of using message queuing for Component Enabler Scalability.

Using MSMQ

There is one Remote Access server service for each installed Runtime environment. This service handles all messaging for any System deployed to that Runtime environment. The MSMQ Remote Access Server service is configured to start automatically when Windows is started. The Runtime administrator is not required to do anything to enable clients to communicate with Systems via MSMQ.

There is an individual MSMQ queue for each configured Remote Access server view. This MSMQ queue is used for all messages sent to the System configured for that view.

Note that by default, the MSMQ Remote Access server service is not installed when Enterprise Application Environment Runtime is installed. To install, the relevant option from the Custom installation dialog must be selected. MSMQ must exist on the Windows server computer before installing the MSMQ Remote Access server service. If it does not exist, the installation procedure for the MSMQ Remote Access server will not be able to continue.

Configuring and Managing Message Queues

Microsoft Message Queues can be created and managed by the Runtime administrator using the Computer Management MMC application.

To display the MMC Computer Management window, select Control Panel/ Administrative Tools/ Computer Management.

The Computer Management application is supplied with the Windows Operating System. When MSMQ is installed as an optional part of Windows, the Message Queuing entry in the Computer Management application window becomes available. Runtime administrators can then create and delete MSMQ queues. Additionally, messages on these MSMQ queues can be viewed and purged, if required.

Runtime uses one MSMQ queue for each configured Remote Access View. The name of this MSMQ queue is the concatenation of the Runtime environment name and the view name. For example, for environment EAE32 and view TPCC_VIEW, the name of the MSMQ queue used will be EAE32.TPCC_VIEW.

The Runtime administrator can use the Computer Management program to pre-create the MSMQ queue required for each view for each Runtime environment. Alternatively, the MSMQ Remote Access service for each Runtime environment will create these MSMQ queues for each of its views when that service is started. However, any MSMQ queues created by the MSMQ Remote Access service are deleted when the service is stopped. Consequently, all messages in these queues are also deleted.

Administering Your Remote Access Server

Starting and Stopping

Use the Services option in the Control Panel to stop and start a server session. The service is shown as *"<environment name> Remote Access (TCP/IP)"* in the Services list.

The MSMQ service is listed as *"<environment name> Remote Access (MSMQ)"*, although it is configured to start automatically.

Logging

You can set your logging options for a View. These options are for automatic logging on or off, or for logging at the client's request. If this last option is specified, the client can initiate tracing in its connection request.

All errors are written to the errors.log file in the %LINCIIDIR%\log directory. When logging is turned on, additional tracing messages are logged to the ratltrace.log in the same directory.

Security

Password aging may be enforced when adding a new user through the Windows User Manager window. You can also control the minimum/maximum password age for a group, or individual user, by choosing Account from the Policies menu on the **User Manager** window (Start > Programs > Administrative Tools (Common) > User Manager).

Timeout (WDP Client)

You can enter the length of time, in seconds, that the WDP client waits for input from the attached Graphical Interface Workbench client on the Workstation Driver Program (WDP) page of the Environment Configuration dialog box in the Admin Client. The default value is 30000 seconds.

Time out (Remote Access Server Client)

You can enter the length of time, in seconds, that the Remote Access Server client waits for input from the attached Component Enabler client on the RATL page of the Environment Configuration dialog box in the Admin Client. Enter 0 for no time out, which is also the default value.

Note: *The service must be restarted before changes to the Timeout will value take affect.*

Configuring the Remote Access Server on UNIX Operating System

Server configuration information is stored in two configuration files located in the $LINCIIDIR/config directory. They contain the following information:

- **ActiveLINC.env** contains the environment information. It is generated as the output of the provided script mkenvfile, and may need some modification, for example if the System is built in interpretive mode.
- **ActiveLINC.cfg** contains general server configuration information and View configuration information. You must create this file before you use the server for Component Enabler.

To configure the server software for Component Enabler, you need to configure the two configuration files, as described in the following:

1. To create ActiveLINC.env, run the script, mkenvfile. The following rules apply to the ActiveLINC.env file entries:
    - Lines in the file have the form "**variableName=value**".
    - Variables should not be defined in terms of other variables, as variable expansion is not supported.
    - Comment lines start with the hash character (#).
2. To create ActiveLINC.cfg, copy $LINCIIDIR/config/alinc.cfg to $LINCIIDIR/config/ActiveLINC.cfg. Edit the server-wide configuration data in the [ActiveLINC] section of the file and create [VIEW] sections as required.

    The following rules apply to the ActiveLINC.cfg file:

    - ActiveLINC.cfg contains sections, each delimited by a label, **[LABEL]**.
    - The first section, labelled **[ActiveLINC]**, contains the general server information. **[ActiveLINC]** denotes the server configuration sections, and cannot be used as a View name.
    - There is a separate section for each View, labelled with the name of the View, **[ViewName]**.
    - The lines in this file contain text in the form "**label=value".**
    - Comment lines can take any form, but should not start with the name of any of the values specified in the following section or contain the equals sign (=).

    A sample ActiveLINC.cfg file is shown below.

    ```
    --------Start of file--------
    [ACTIVELINC]
    PublicPassword=activelinc
    LoggingChannel=0
    WDPListenerPort=6004
    # LINC language mappings
    ENGLISH=04ISOL02EN04ISOC02US
    FRENCH=04ISOL02FR04ISOC02CA
    GERMAN=04ISOL02DE04ISOC02DE

    # Two sample views have been created below.
    ```

```
# Please alter the views to suit your own system.

[VIEW1NAME]
SystemName=SAMPLE
SystemDirectory=/u1/linc/SAMPLE
UsernameLabel=Userid
UsernameGreeting=Enter your Userid
PasswordLabel=password
PasswordGreeting=Enter your password
OracleSID=JO
# Prevent access to users fred and billy
ForbiddenUsers=fred billy
# Prevent access to users in the 123.123.123. domain
ForbiddenIPs=123.123.123.*

[VIEW2NAME]
SystemName=SAMPLE
SystemDirectory=/u1/lincrus/SAMPLE
AutomaticUser=alinc001:alinc
OracleSID=PF
--------End of file--------
```

Creating Views

Views provide mapping and security information for users to connect to a remote System. Views contain attributes that are predefined for an application. For example, a View can specify whether users are required to login using a user id and password or whether they can access the application as anonymous users. It also contains mapping information, so that remote users do not have to know system settings, such as the ORACLE SID or the OS 2200 Transaction Code.

The following table describes the labels used in the ActiveLINC.cfg file for configuring Component Enabler Views for a server on a UNIX platform.

Name	Section	Mandatory	Description
PublicPassword	ActiveLINC	Yes	The password of the user id, ALPublic. For Views with automatic login, the process first runs as the specified AutomaticUser and later as user ALPublic. This is required by a current limitation in Enterprise Application Runtime.
LoggingChannel	ActiveLINC	No	Logging in Component Enabler uses syslog. Syslog provides eight channels for user applications. Valid settings for LoggingChannel are 0 – 7. An invalid or absent setting will result in channel 0 being chosen.

Name	Section	Mandatory	Description
WDPListenerPort	ActiveLINC	Yes	The Remote Access server contains a subset of the WDP listener functionality. As Component Enabler clients do not have their own WDP process running, this setting is necessary to allow them to download items from the WDP repository. The value specified, in the range 1001-65535, must not clash with port numbers currently in use by other processes on the System, especially the server port.
{Language}	ActiveLINC	No	Provides a mapping between a language and the ISO codes that define the language. A language mapping is in the format <language name >=04ISOL02XX04ISOC02YY, where XX is the ISOLANGUAGE abbreviation and YY is the ISOCOUNTRY abbreviation.
SystemName	ViewName	Yes	The name of the System that this View references.
SystemDirectory	ViewName	Yes	The absolute path to the system directory.
OracleSID	ViewName	Yes	The ORACLE SID that the System is using.
UsernameLabel	ViewName	No	A label that appears beside the Username field in the login screen.
UsernameGreeting	ViewName	No	A message that appears above the Username prompt in the login screen.
PasswordLabel	ViewName	No	A label that appears beside the Password field in the login screen.
PasswordGreeting	ViewName	No	A message that appears above the Password prompt in the login screen.
AutomaticUser	ViewName	No	The user id and password for the Automatic User. This information is used to log users in when they request a View that does not require a login. This must be in the form: {userid}:{password}.
ForbiddenUsers	ViewName	No	A space-separated list of users who will be denied access to the View. If this is set, only users who are not listed will be granted access.
PermittedUsers	ViewName	No	A space-separated list of users who will be permitted access to the View. If this is set, only users who are listed will be granted access.
ForbiddenIPs*	ViewName	No	A space-separated list of IP addresses from which access to the View will be denied. If this is set, only users from an IP address that is not listed will be granted access.

Name	Section	Mandatory	Description
PermittedIPs*	ViewName	No	A space-separated list of IP addresses from which access to the View will be permitted. If this is set, only users from an IP address that is listed will be granted access.
Logging	ViewName	No	Specifies the logging condition. • 0 logging is set on at the client's request • 1 logging is forced on • 2 logging is forced off.

Notes:

1. *The values marked with an asterisk (*) have limited support for wildcards. Each IP address within the string must be of the form a.b.c.d where a,b,c,d is a value; for example, 192.146.252.74. When using wildcards, the valid format is 192.146.*.**
2. *Forbidden and permitted IP addresses are mutually exclusive.*
3. *The Oracle SID must be specified.*
4. *The Automatic login will be enabled in Enterprise Application Runtime for UNIX operating systems through EIFs for each platform. Once enabled, the ALINTER process will connect to a user System using the ALPublic user id. ALPublic must be set up as a UNIX user id and the password that is set must also be added to the [ActiveLINC] section of the ActiveLINC.cfg file.*

Using Anonymous Login

To enable anonymous login you must set the following options in the ActiveLINC.cfg file:

```
[ActiveLINC]
PublicPassword=<password>
[ViewName]
AutomaticUser=<usercode>:<password>
```

Setting these options allows you to automatically log in to a View without entering a usercode or password.

If you initiate additional anonymous login sessions to the View they will all share the same GLB.WORK area. This means that each user’s data will overwrite the data of the other users. To avoid this problem add the following line to $LINCIIDIR/cinfig/linc.rc:

```
LINCII_ENABLE_ANONYMOUS=Yes
```

Setting the Timeout Threshold

The internal timeout threshold of the Remote Access server can be set using the variable LINCII_ALINCTMOUT. This sets the time to wait between server responses in seconds. The default value is one second.

To set the timeout threshold, enter the following in the ActiveLINC.env file:

```
LINCII_ALINCTMOUT= <number of seconds>
```

Message Queuing

Message Queuing allows multiple concurrent users of Component Enabler to access an Enterprise Application Environment System, by making more efficient use of system resources, and reducing the overheads associated with establishing and maintaining individual connections for each user session. This is an alternative to the existing method of connecting to the Runtime System using individual TCP/IP connections.

See your *Unisys Enterprise Application Component Enabler Developer's Guide* for further details of using message queuing for Component Enabler Scalability.

The following table describes the labels specific to message queuing that are used in the ActiveLINC.cfg file:

Name	Section	Mandatory	Description
QueuePath	ActiveLINC	Yes	The path name to the message queue formatted for use by the Geneva MQ Client. See your Geneva MQ documentation for further information.
AdminPort	ActiveLINC	Yes	The Remote Server for message queues runs as a daemon
MaxQueueServers	ActiveLINC	No	The number of servers started to service this view.

Processing

The QueuePath is used to open the central request queue. The path name consists of a Message Queue Server name and a path on that server which identifies the queue. The queue name can be either private or public.

For example, private queue "EAEQ" on server MQSRV:

```
MQSRV\$PRIVATE\EAEQ
```

For example, public queue "EAEQ" on the default MQ server:

```
.\EAEQ
```

The server name and default server are configured when you setup the Geneva MQ software.

On startup, the *AdminPort* will be monitored by a java process called CEMonitor. MaxQueueServers ALINTER processes will be started for each view where MaxQueueServers has been specified. There will also be a queue created and opened on the server hosting the MSMQ indicated by the server name in the QueuePath setting.

Administering Your Remote Access Server

Starting and Stopping Standard Connections

To start and stop the server, you must become the root user.

To dynamically start the server on port 2449, the default port for servers, use the following command:

```
alincsrv -p 2449 &
```

Note: *Ensure that the WDPListenerPort is not set to the same port as the server.*

To automatically start the server, add the following command to the script that is in the $LINCIIDIR/sample directory and copy the script to the system boot directory:

```
alincsrv -p <port number>
```

To stop the server, you must determine the pid of the server, then kill the process, using the -15 option, to ensure that an orderly shutdown occurs. The following example shows these commands:

```
ps -fu root | grep alincsrv
kill -15 <pid of alincsrv>
```

Message Queuing Starting and Stopping

Use the **rascfg** command to start and stop the Remote Access server when message queuing is to be used. Before executing the **rascfg** command the Java CLASSPATH environment variable must contain the Geneva MQ java class archive GMQ.jar.

To start the server use the following syntax, where GMQDIR is the directory in which the GMQ java client is installed:

```
export CLASSPATH=$GMQDIR/GMQ.jar
rascfg start
```

To stop the server use the following syntax, where GMQDIR is the directory in which the GMQ java client is installed:

```
export CLASSPATH=$GMQDIR/GMQ.jar
rascfg stop
```

Security

To enable password aging when starting the service for a server, enter the following command:

```
alincsrv -a
```

The switch *-a* instructs the Remote Access server to use the system password settings. This enables the following security measures:

- All users who have been allocated a system-defined password expiry date, will not be able to log in after this date has passed.
- All users who have been given a password age limit, will be informed on login that the password is older than the time limit, and that they need to supply a new one.

  A newly supplied password must comply with the system specific requirements of the *passwd* command, regarding difference from the old password and the number of characters.

Tracing

Tracing can be started by a client when making the connection request or by the operator. Tracing is performed by the *syslog* daemon. To customize logging, you can specify a syslog channel in ActiveLINC.cfg and configure the channel in the syslog.conf file.

As an example of the tracing capabilities, adding the following line to /etc/syslog.conf sends errors, warnings, and notices on logging channel 0 to the file, /var/log/ActiveLINClog:

```
local0.err;local0.warning;local0.notice /var/log/ActiveLINClog
```

Note: *You will have to get* ***`syslogd`*** *to re-read the configuration file after any changes. Starting and stopping* ***`syslogd`*** *will achieve this.*

Tuning the PC to Remote Access Server Connection Timeout Parameters

When the connection between the PC and the Remote Access Server breaks on the PC side, the Remote Access Server should be able to understand this and close the session.

For this purpose Remote Access Server uses the SO_KEEPALIVE option. This means that after a crash of the client, the connection will time out on the server side in 120 or 130 minutes (depending on the state of the client computer - working or not). There are kernel parameters which allow you to tune these times.

For a TCP/IP connection there are parameters/options, which can not be set for a particular connection/socket - as they are set for ALL TCP/IP connections on the host computer - such as the kernel parameters TCP_KEEPIDLE, TCP_KEEPINTVL, which define the frequency of the probes sent by a SO_KEEPALIVE socket. The number and the names of these kernel parameters are different on the different UNIX platforms.

- On Sun, the time can be tuned using

  ```
  ndd -set /dev/tcp tcp_keepalive_interval mmmmmm
  ```

  where *mmmmmm* is time in milliseconds.
- On UnixWare, the following can be tuned:

  tcp_initial_timeout, tcp_keepalive_port, tcp_keepidle, tcp_keepintvl

- On PTX,

  TCP_KEEPCNT, TCP_KEEPIDLE, TCP_KEEPINIT, TCP_KEEPINTV

  can be tuned using the PTX/menu admin system.
- On AIX, the time can be tuned using

  ```
  no -o tcp_keepidle=hhhhhh
  no -o tcp_keepinit=hhhhhh
  ```

  where *hhhhhh* is time measured in half seconds.

Configuring the Remote Access Server on MCP

CCF Configuration

Use the CCF components to configure the server entities as follows:

CCF Component	Entity
Router	PCM
CUCI	Device and Service
TCPIP	Port
RATL	Service, View, Language, MQserver, MQclient, and MQrequest

Router

Identify the server as a PCM to the router and enable it, using the following syntax:

```
Add PCM <name>
       codefile = <filepath>
Enable PCM <name>
```

Attributes	Description
<name>	Identifies a valid server name.
<file path>	Specifies the codefile path.

Example:

```
Add PCM ratlpcm
       codefile = system/linc/pcm/ratl
Enable PCM ratlpcm
```

CUCI

As part of the CCF configuration you must configure a:

- Connection Device
- Connection Service

Connection Device

Configure a device to the CUCI PCM to associate device connection attributes with each dialog. Attributes are assigned using the following syntax:

```
Add Device <device name>
       acvt = <VTname>,
       ccenable = <boolean>,
       controlcapable = <boolean>,
       dynamic = <boolean>,
       marccapable = <boolean>,
       maxinput = <number>,
       maxoutput = <number>,
       messages = <category>,
       securitycatlist = <name>,
       ndlheader = <boolean>,
       screen = <boolean>,
       usage = <category>
```

Attributes	Description
<device name>	Defines a unique device name.
acvt	Specifies an Application Controller Virtual Terminal. This is the virtual terminal COMS expects the client to be using on input. This attribute is optional. <VT name> must be set to either Transparent or Default.
ccenable	Indicates if control character processing is enabled. This attribute is optional. This attribute can only be set to False.
controlcapable	Indicates if control commands can be entered or not. This attribute is optional. This attribute can only be set to False.
dynamic	Indicates if the connection is to be kept by COMS when the connection terminates. This attribute is optional.
marccapable	Indicates if the connection can handle screen output from MARC. This attribute must be set to True if you are to use the RATL Services attribute **MARCOpenText**.
maxinput	Specifies the maximum number of bytes in an input message. For Big Buffer Ispecs, this must be set to 45300, or 16100 if using Graphical Interface Workbench. <number> must be a minimum of 2500.

Attributes	Description
maxoutput	Specifies the maximum number of bytes in an output message. For Big Buffer Ispecs, this must be set to 45300, or 16100 if using Graphical Interface Workbench. <number> must be a minimum of 2500.
messages	Determines whether system messages are displayed at the client. <category> must be set to None.
securitycatlist	Specifies a COMS Security Category List configuration entity that is defined in COMS. When set, the value of this attribute is the default value used when connecting to COMS. It can be overridden by the service attributes of NOFidentity and/or PCEidentity. This attribute is optional.
ndlheader	This attribute is optional. This attribute can only be set to False.
screen	Indicates whether the client is a screen device. This attribute is optional. This attribute can only be set to False.
usage	Identifies whether the client can receive input messages, send output messages, or both. <category> must be set to one of In, Out, or IO. The recommended value is IO, so that the client can both send and receive messages.

Example:

```
Add Device RATLweb
       acvt = transparent,
       ccenable = false,
       controlcapable = false,
       dynamic = true,
       marccapable = false,
       maxinput = 10000,
       maxoutput = 10000,
       messages = none,
       securitycatlist = SCL_NOF,
       ndlheader = false,
       screen = false,
       usage = IO
```

Connection Service

Configure a service to the CUCI PCM to associate device connection attributes and to identify the connection path for each dialog. Attributes can be assigned using the following syntax:

```
Add Service RATLon
       closeaction = <number>,
       device = <name>,
       logoffdisconnect = <boolean>,
       dynamic = <boolean>
Enable Service RATLon
```

Attribute	Description
closeaction	The SYSTEM/COMS Close Action as defined on the SYSTEM/COMS Usercode menu. The close action is also defined on the Station menu of the COMS Utility. You must enter a value in the range 1 to 4. ***Note:*** *If the* dynamic *attribute is set to True the action is that of the DefaultStation definition in COMS.*
device	You must provide a CUCI Device Name. This does not need to be a valid SYSTEM/COMS Device name.
dynamic	Indicates if the connection is to be kept by COMS. This attribute is optional. Possible values: • True, the connection will not be kept by COMS • False, the connection will be kept by COMS.
logoffdisconnect	Indicates whether the session is automatically logged off when the connection is terminated. This attribute must be set to True.

Example:

```
Add Service RATLon
       closeaction = 3,
       device = ratlweb,
       logoffdisconnect = true
Enable Service RATLon
```

Note: *For services, the* **`service name`** *associated with the declared service is the name of the next service in the connection path.*

TCPIP Port

Define a port to the TCPIP PCM to associate port attributes for the dialogs. Attributes can be assigned using the following syntax:

```
Add Port <port name>
       stationname = <name>,
       checkinterval = <number>,
       device = <name>,
        framing = <category>,
        maxoffer = <number>,
        maxoutput = <number>,
        service = <name>,
        socket = <number>,
        transport = <category>,
        windowsize = <number>
Enable Port <port name>
```

Attribute	Description
<port name>	Specifies a unique port name. <port name> must correspond with an identified service.
stationname	Specifies a unique station name. See "Using StationName" below.
checkinterval	Specifies the period of inactivity before "keep alive" packets are sent. <number> is specified in seconds, and must be in the range 0 through 1440.
device	Specifies the name of a device to be used for all subport connections that are passed to the CUCI PCM. <name> must be a previously configured device.
framing	Specifies the type of message delineation to be used. Message delineation indicates where a message starts and ends. <category> must be set to Standard.
maxoffer	Specifies the number of subports to be offered (made available for connection) at any time. <number> must be in the range 0 through 31.
maxoutput	Specifies the maximum number of bytes in an output message. For Big Buffer Ispecs, this must be set to 45300, or 16100 if using Graphical Interface Workbench. <number> must be in the range 128 through 65535.
service	Specifies the name of a service to be used for inbound subport connections. This is the next service in the connection path. <name> defaults to <port name>.
socket	Specifies the socket number used by the client to connect to the port file. <number> should be set to the recommended value of 2449, unless a non-standard port configuration exists.
transport	Identifies the transport to be used. <category> must be set to TCPIP, or it will default to null.
windowsize	Specifies the maximum number of bytes that can be queued, on input, per dialog before more is received. For Big Buffer Ispecs, this must be set to 45300, or 16100 if using Graphical Interface Workbench.

Example:

```
Add Port RATL
        stationname = actlinc/#,
        checkinterval = 5,
        device = ratlweb,
        framing = standard,
        maxoffer = 1,
        maxoutput = 10000,
        service = ratl,
        socket = 2449,
        transport = tcpip,
        maxoffer = 500
Enable Port RATL
```

Using StationName

This attribute names a connection to COMS. It provides a flexible format from which station names can be generated.

Note: *See the <tcpip psnf> field of the Add or Modify TCPIP PCM command in the* ClearPath HMP Series CCF Administration Guide *for all possible stationname values.*

When developing a station name you need to consider uniqueness and determinability. For example, you may require that the station name be consistent each time a particular user connects, and will not change over time. You may also require that more than one connection be used from the same client or from any number of clients and each name used is unique.

You also need to consider any COMS security that is applied to station names and how these stations comply.

For the best means of ensuring a consistently unique name, use the <tcpip psnf> attributes $yourhost or $yourIPaddress within the stationname attribute value.

If a hostname is desired but an IP address is formed for the stationname instead, that is the letters “IP” are inserted in front of the address, it may be possible to use TCP/IP mapping to set the hostname. For details on mapping IP addresses, see your *TCP/IP Implementation and Operations Guide*.

On some networks, such as Dynamic Host Configuration Protocol (DHCP), IP addresses are not consistent over time. In this situation there are two possible alternatives:

- Reserving a period of time where the IP address will remain stable.
- Configuring a Domain Name Server (DNS) so TCP/IP can resolve an IP address to a known hostname.

  For details on configuring a DNS Resolver, see your *TCP/IP Distributed System Services (DSS) Operations Guide*.

The effect of StationName on station name (GLB.STN)

The setting of the internal station name, or System Data Item GLB.STN, is affected by the external (COMS) station name.

For Component Enabler (or NOF) based connections the value of GLB.STN is based on the COMS station name value and the Remote Access service attribute StationNamePrefix value. If the StationNamePrefix is not defined then GLB.STN is prefixed by the letters “RAT”, as shown in the first example below. If the derived station name is longer than the limit of 17 characters, the connection attributes for **yourhost** is used, as shown in the second example below. If the value for yourhost is not known or it is longer than the 17 character limit, the value for **youripaddress** is used. If this value is also too long the IP address is converted to its integer equivalent, as shown in the third example below.

The following example shows the resulting COMS and station names for a client with the specified Port Stationname format, a station name prefix set or not set to “SNP”, a hostname of Timbertown, and an IP address of 123.132.213.231:

TCPIP Port StationName	RATL Service StationName Prefix	COMS Station Name	GLB.STN
ACTLINC/#	<null>	ACTLINC/1	RATACTLINC/1
	SNP	SNPACTLINC/1	ACTLINC/1
$yourhost/$youraddress	<null>	TIMBERTOWN/1234	RATTIMBERTOWN/1
	SNP	SNPTIMBERTOWN/1234	TIMBERTOWN/1
$youripaddress/#	<null>	123_132_213_231/1	RAT2072303079/1
	SNP	SNP123_132_213_231/1	123_132_213_231/1

For Graphical Interface Workbench based connections the value of GLB.STN is set using the station name followed by the COMS station name and prefixed by the letters “WDP”. If this format exceeds the 17 character limit “WDP/” is used followed by the COMS Station designator's index value.

RATL Configuration

Configure the following entities:

- Service
- View
- Language

RATL Configuration Syntax

You can use the following verbs in CCF to configure the server for Component Enabler, Graphical Interface Workbench Enterprise Application Workbench, or Web Enabler:

Verb	How it is Used
Add	To define attributes for Views, Languages, Services, MQservers, MQclients, and MQrequests.
Delete	To change existing Views, Languages, Services, MQservers, MQclients, and MQrequests. The View, Language, Service, MQserver, MQclient, or MQrequest must be disabled before it can be deleted or modified.

Verb	How it is Used
Disable	To change existing Views, Services, and MQrequests. The View, Service, or MQrequest must be disabled before it can be deleted or modified.
Enable	To enable existing Views, Services, and MQrequests.
List	To list Dialogs, Services, Languages, Views, MQservers, MQclients, and MQrequests.
Modify	To define attributes for Views, Languages, Services, MQservers, MQclients, and MQrequests. The View, Language, Service, MQserver, MQclient, and MQrequest must be disabled before it can be deleted or modified.
Option	To change server program runtime options.
Show	To show existing Views, Services, Languages, connected Dialogs, MQservers, MQclients, and MQrequests.
Status	To show the status of the PCM, such as compile time, code version, and title.
Trace	To alter or show the trace options.

Service

Identify a service to the PCM to associate the connection path for dialogs. Attributes are assigned using the following syntax:

```
Add Service <service name>
       service = <name>,
       NOFidentity = <name>,
       PCEidentity = <name>,
       SwitchToFireUp = <boolean>,
       StationNamePrefix = <name>
Enable Service <service name>
```

Attribute	Description
<service name>	Defines a unique service name.
service	Specifies the name of a service to which inbound stations should be connected. This is the next service in the connection path. <name> must be a previously configured service.
NOFidentity	Identifies the connection types of the applications. <name> is a COMS Security Category List configuration entity that is defined in COMS. If the attribute is not configured in the service definition, it must be defined using the SecurityCatList attribute in the CUCI service definition before the server can handle the Component Enabler connections.

Attribute	Description
PCEidentity	Identifies the connection types of the applications. <name> is a COMS Security Category List configuration entity that is defined in COMS. If the attribute is not configured in the service definition, it must be defined using the SecurityCatList attribute in the CUCI service definition before the server can handle the Graphical Interface Workbench connections.
SwitchToFireUp	Specifies behavior when switching back to an application from which a switch previously occurred. This attribute is optional. If True, returns to the Fireup Ispec instead of the Ispec from which the orginal switch occurred. If False, returns to the Ispec from which the original switch occurred. If the client has not previously been displayed, the Fireup Ispec is retrieved. Any data contained in the SWITCH.TO command overrides the setting of this attribute.
StationNamePrefix	Prevents the "RAT" station name prefix appearing in GLB.STN for Component Enabler connections. <name> will prefix the COMS Station Name. See “Using StationName”. This attribute is optional. This option should not be set for Graphical Interface Workbench connections.
MARCOpenText	Allows input to be passed to the MARC dialog when a new COMS connection is established from the server to a desired application. The value must be enclosed in double quotation marks. When this atttribute is present the text is sent instead of the server passing a hard coded message to the MDPLAUNCH window. ***Note:*** *In order for the command to be carried out successfully by MARC, the CUCI Device attribute* ***`marccapble`*** *must be set to true.*

Example:

```
Add Service RATL
      service = LINC,
      NOFidentity = SCL_NOF,
      PCEidentity = SCL_PCE,
      SwitchToFireUp = False,
      StationNamePrefix = AL/
Enable Service RATL
```

Configuring Remote Access Server Entities

The Remote Access configuration entities for the COMS Security Category List names must be configured in **at least one** of the following attributes of the CCF configuration:

- SecurityCatList in the CUCI service definition

- NOFidentity and/or PCEidentity in the service definition

See “RATL Configuration Syntax” for a description of the syntax used to configure the server in an MCP environment.

When configuring entities you can change their names, but the names must remain unique. You can define additional Port and Service entities ensuring that a connection path can be resolved through CCF, encompassing the PCMs from TCIP using the Remote Access server to CUCI. See the *ClearPath HMP NX/Services Administration Guide* or *ClearPath HMP Series CCF Administration Guide* for information on using CCF.

View

You can identify Views to the PCM to associate connection criteria for Component Enabler users to access applications. A View can refer to one or more applications, or different Views may apply to the same application. To identify at least one application, you can assign attributes using the following syntax:

```
Add View <view name>
       application = <system>,
       level = <number>,
       window = <name>,
       usercode = <usercode>,
       language = <language>
Enable View <view name>
```

Attribute	Description
<view name>	Specifies a unique view name.
application	Indicates the name of the initial System to be accessed through this View. This attribute is optional. <system> may contain the usercode, the System name and the Dictionary pack name of the generated System. The Dictionary pack contains the LINCGLI, LINCFORM, and LINCCNTL files for the generated System.
level	Indicates the command level for dialogs using this View. This attribute is optional and only applies to NOF based connections.
window	Specifies the COMS window to be used by this View. This attribute is optional, but is required when the application to which you may need to switch is not already in use.
usercode	Indicates the usercode for every dialog that is established for this View. If it is not present, the client is asked for a usercode. This attribute is optional. <usercode> must be valid for the purposes of workstations connected to Runtime through COMS.
language	Indicates the preferred language to be used for this View. This attribute is optional. <language> must be a previously identified language.

Example:

```
Add View sample_auto
       application = (USER1)SAMPSYS on DICTPACK,
       level = 0,
       window = SAMPLE,
       usercode = anonymous,
       language = english
Enable View sample_auto
```

Language

Identify a language to associate the ISO Country code and an ISO Language code with a language name. The ISO Language code is defined in ISO 639 while the ISO Country code is defined in ISO 3166. These standards are available in the official ISO website, http://www.iso.ch/.

Language names are configured in the System. Attributes can be assigned and enabled using the following syntax:

```
Add Language <language>
       ISOC = <country code>,
       ISOL = <language code>
```

Attribute	Description
ISOC	Specifies the ISO Country code to be associated with the language name.
ISOL	Specifies the ISO Language code to be associated with the language name.

Example:

```
Add Language english
       ISOC = EN,
       ISOL = EN
```

Message Queuing

Message Queuing allows multiple concurrent users of Component Enabler to access an Enterprise Application Environment System, by making more efficient use of system resources, and reducing the overheads associated with establishing and maintaining individual connections for each user session. This is an alternative to the existing method of connecting to the Runtime System using individual TCP/IP connections.

See your *Unisys Enterprise Application Component Enabler Developer's Guide* for further details of using message queuing for Component Enabler Scalability.

The following entities are added to the PCM section of the CCF params file to configure the request queues and connection details of the FalconMQ Server and FalconMQ Client library:

- MQServer
- MQclient
- MQrequest

MQServer

```
Add MQServer <name>
       Servername = <name>,
       IPaddress = <ipaddress>,
       Port = <number>,
       Domain = <name>,
       UserName = <name>,
       Password = <name>,
```

Set attributes associated with the FalconMQ server using the following syntax:

Attribute	Description
Servername	The name of the FalconMQ server. Default: <MQserver name>
IPaddress	The IPaddress of the FalconMQ server. An assigned IPaddress overrides any specified Servername.
Port	The required port number. Default: 0
Domain	The Windows security domain. Default: nulls
UserName	The Windows user name. Default: nulls
Password	The password associated with the Windows user name. Default: nulls

Note: *Case sensitive names that refer to a Windows entity can be enclosed by single quotes to prevent uppercasing.*

You can also use the following commands:

- Modify attributes of existing server entities

  ```
  modify MQServer <name>
           <attribute> = <value>,
  ```

- Itemize all declared MQServers

  ```
  list MQServers
  ```

- Display the attributes for the selected server

  ```
  show MQServer <name>
  ```

- Remove the selected server entity

  ```
  delete MQServer <name or list>
  ```

Example (*italics* denote responses):

```
Add MQserver FMQS
       IPaddress = 123.1.2.3,
       Port = 1100,
       Domain = realm,
```

```
       UserName = dnote,
       Password = test,
MQserver FMQS added

Show MQserver FMQS
1 FMQS
   IPaddress = 123.1.2.3
   Port = 1100
   Domain = realm
   UserName = dnote
   Password = test
```

MQclient

Set attributes associated with the FalconMQ client using the following syntax:

```
Add MQclient <name>
       Functionname = <name>
```

Attribute	Description
Functionname	The System Library function name. Default: <MQclient name>

You can also use the following commands:

- Modify attributes of existing client entities.

  ```
  modify MQclient <name>
         <attribute> = <value>,
  ```

- Itemize all declared MQclients.

  ```
  list MQclient
  ```

- Display the attributes for the selected client.

  ```
  show MQclient <name>
  ```

- Remove the selected client entity.

  ```
  delete MQclient <name or list>
  ```

Example (*italics* denote responses):

```
Add MQclient FalconMQSupport
MQclient FALCONMQSUPPORT added

Enable MQclient FalconMQSupport
MQclient FALCONMQSUPPORT enabled
```

MQrequest

Set attributes associated with the request queue using the following syntax:

```
Add MQrequest <name>
       MQclient = <name>,
       MQserver = <name>,
       Pathname = <name>,
       Processes = <number>,
       StationName = <SNformat>,
       Usercode = <identifier>,
       View = <name>,
       Service = <name>
```

Attribute	Description
MQclient	The name of the FalconMQclient.
MQserver	The nameof the FalconMQ server.
Pathname	The request queue path anme. Default: .\PRIVATE$\<MQrequestname>
Processes	The number of queue reader processes. Default: 1
Stationname	The pooling station name format. The possible Snformats are as follows: • <literal> • / • # • $ – MQclient – MQrequest – MQserver – ServerName – IPaddress – Domain – Username Default: $MQrequest/#
Usercode	The pooling user code.
View	The name of the view. Default: <MQrequestname>
Service	The service name. Default: <first service name defined(in Remote Access Server)>

You can also use the following commands:

- Modify attributes of existing MQrequest entities.

```
modify MQrequest <name>
       <attribute> = <value>,
```

- Terminate the selected MQrequest reader process.

```
disable MQclient <name>
```

- Execute the selected MQrequest reader process.

```
enable MQrequest <name>
```

- Itemize all declared MSQclients.

```
list MQrequest
```

- Display the attributes for the selected client.

```
show MQclient <name>
```

- Remove the selected MQrequest entity.

```
delete MQrequest <name or list>
```

Example (*italics* denote responses):

```
Add MQrequest sampleQ
       MQclient = FalconMQSupport,
       MQserver = FMQS,
       Usercode = legion,
       View = samplesystem,
       Service = ratl
MQrepuest sampleQ added

List MQrequests
MQrequests:
1 sampleQ
```

COMS Configuration

To define connection types to applications, you need to define two COMS configuration entities. These are Security Category List and Installation Data entities. The names used in the CCF configuration for SecurityCatList (CUCI device attribute), NOFidentity, and PCEidentity (service attributes) should correspond to the names of Security Category List entities in the COMS configuration. Each Security Category List entity needs a corresponding Installation Data entity. The value assigned to the INTEGER1 attribute determines the type of connection for that entity. The following table shows the allowable values of INTEGER1 for servers.

INTEGER1 Value	Type of Connection
8	Enterprise Application Workbench or Web Enabler (WDP) connections
9	Component Enabler (NOF) based connections

Enterprise Application Workbench and Web Enabler (WDP) Connection Types

You can define Enterprise Application Workbench or Web Enabler connection types for applications using the following COMS configuration entries:

```
CREATE INSTALLATION_DATA RATL_PCE
       INTEGER1 = 8
CREATE SECURITY_CATEGORY_LIST RATL_PCE
       ID = RATL_PCE
```

Component Enabler (NOF) Based Connection Types

You can define Component Enabler based connection types for applications using the following COMS configuration entries:

```
CREATE INSTALLATION_DATA RATL_NOF
       INTEGER1 = 9
CREATE SECURITY_CATEGORY_LIST RATL_NOF
       ID = RATL_NOF
```

Remote Access Server Commands

Clear Command

This command is used to clear connections from the PCM.

Syntax:

```
Clear Dialog<#,name, or list>
```

Example:

```
Clear Dialogs 1,4-7,9
```

Option Command

The following list outlines the options users can set:

- AllAttributes
- ShowAsserts
- AssertDump
- CompactTables
- LogicError1
- LogicError2
- ForwardSyncMessages

  When set any received CCF Sync protocol messages are sent onwards, otherwise they are ignored.

- ObfuscateLoginResponses

  When set allows Graphical Interface Workbench sessions to receive Remote Access server Login Reponse messages in obfuscated form. Existing versions of Graphical Interface Workbench prior to 3R1 do not expect Login Response messages in obfuscated form.

Syntax:

```
Option [+/-] <option list>
```

Example:

```
Option +ObfuscateLoginResponses
```

Administering Your Remote Access Server

Starting and Stopping

Use the CCF commands to start and stop the server and to monitor or trace server activities.

Note: *Any connections to COMS or a COMS application via NX Services (for example, NX view) are terminated.*

Because the server is configured as part of CCF, starting and stopping CCF will start and stop the server:

- To start Remote Access Server along with CCF, use:

  ```
  NA CCF+
  ```

- To stop Remote Access Server along with CCF, use:

  ```
  NA CCF-
  ```

Alternatively the server can be started and stopped without disrupting CCF:

- To start Remote Access Server without affecting CCF, use:

  ```
  NA CCF enable pcm ratlpcm
  ```

- To stop Remote Access Server without affecting CCF, use:

  ```
  NA CCF disable pcm ratlpcm
  ```

Monitoring

You can monitor the status of any Remote Access server component configured in CCF, such as Languages, Views, and Dialogs. You can use the CCF **STATUS** or **SHOW** commands to display the status of the selected component.

The following example uses the **STATUS** command to display the status of the PCM. It displays the version, timestamp, internal program options and default language.

```
NA CCF RATLPCM STATUS
```

This command returns the following data:

```
Unisys Corporation COMSock
RATLPCM 44.201.1 -09/20/98 - 10:29:31
Compile Options:Trace, Assert
Language = ENGLISH
```

The following example uses the **SHOW** command to display the status of the connected dialog.

```
NA CCF RATLPCM SHOW DIALOG 1
```

This command returns the following data:

```
Sunday 09/20/98 10:29:31
1 RATL/UNO
Client = Open
Transport = Open
Service = Open
Input Seq Num = 8
Output Seq Num = 6
NOF = True
GUI = False
View = UNO
Language = SPANISH
Device = RATLWEB
Maxoutput = 6000
HostName = ACUSAHA
ACVT = Default
```

Security

If a host System supports password aging and the Remote Access server is compliant with the System then user passwords can be changed when using the Remote Access server.

For more information about setting password aging on an A Series host, see your *MCP/AS Security Administration Guide.*

Tracing

Use CCF commands to start and stop tracing and to set the tracing attributes. Tracing can be initiated either by the operator, or by the client as part of the connection request. Tracing is normally used only for problem resolution, as it may impact the performance of the server.

The options for the CCF TRACE command are:

ON or RESUME	Turns tracing on.
OFF or SUSPEND	Turns tracing off.
CLOSE	Closes the current trace file.

Users can set trace attributes to specify which activity will be traced. The trace attributes are listed in the following table:

Attribute	Components Traced
RATL	RATL protocol
ALLDLGS	All dialogs
ATTACH	Session establishment and termination
Blocked	Suspended and resumed output
CMDINFO	CCF command data buffer
CREDITS	Bytes available for transmission
DATAINFO	CCF associated data buffer
DATAPATH	Procedural flow of message buffering
FULL	Shows full size of data buffers
LISTBUFS	Internal data space processing
LOCKS	Contention
MISC	Miscellaneous
MISCPROCS	Miscellaneous processing
MQ	Message Queue interface
MSG	Message displays
OPERINP	Operational interface
PFD	Configuration file parameters
SCANNER	Parsing
TBL	Tables

An example of the CCF command to start tracing all dialogs for the Remote Access server is as follows:

```
NA CCF RATLPCM TRACE ON +ALLDLGS
```

Remote Access Server Protocol Response Messages

In particular situations the server sends protocol messages, with a number of possible response codes, to the client.

The following table outlines some of the more common codes and possible causes, to assist in determining the cause of the message:

Response Code	Meaning	Possible Causes
100	Successful operation	Session was successfully established.
101	Login is required	The View does not contain a usercode attribute setting.
103	Additional login	The userdata file indicates that other required user related attributes are needed.
201	System does not exist	Either the View is not known to the server or the application referred to in the View does not exist.
203	System cannot be contacted	Unable to read or obtain the desired information from the control file.
204	Access denied	• System is not DW capable • System is not 16.3 or later release • Userdata validation error (invalid usercode/ password) • Duplicate connection name • Usercode specified on View does not exist or is not valid • COMS Window for the Application does not exist or is not valid (window list) • Security Category List used by connection does not exist in COMS • Failed COMS Usercode/Station name security checking

Localization of Messages

Remote Access Server messages can be translated using the MCP-based Multilingual System (MLS). The user messages are stored in the program SYSTEM/LINC/PCM/RATL in the following arrays:

- pcm_msgs
- login_prompts
- login_labels

Configuring the Remote Access Server on OS 2200

To configure the server, the System Administrator uses a text editor to create an element called LINC*RATL-SERVERS.CONFIG. There is only one configuration file for all the servers on the System.

Two examples of LINC*RATL-SERVERS.CONFIG elements follow:

```
SERVER NAME,SRAT1 PORT,4016 TSAM-BDI,0200561
SERVER NAME,SRAT2 PORT,4017
SRAT1   TSU,TSURS1  TSU-PWD,TSURS1PWD APP,3 APP,7 APP,8
SRAT2   TSU,TSURS2  TSU-PWD,TSURS2PWD APP,8
PID-INFO  PID-SERVER,SRAT1  APP,8  START-PID,5000  NUM-PIDS,2;  LINC-STATUS,UP
PID-INFO  PID-SERVER,SRAT2  APP,8  START-PID,5101  NUM-PIDS,2;  LINC-STATUS,DOWN
PID-INFO  PID-SERVER,SRAT1  APP,7  START-PID,6000  NUM-PIDS,10;  LINC-STATUS,UP
PID-INFO  PID-SERVER,SRAT1  APP,3  START-PID,7000  NUM-PIDS,10;
     LINC-STATUS,SUSPENDED
VIEW ALIAS,NULVIEW  LINC-SYSTEM,JAL63  STATUS,DOWN  ACCESS,OPEN ;
                       DEFAULT-USERID,MIKE  DEFAULT-PWD,SECRET
VIEW ALIAS,YYYYY  LINC-SYSTEM,TST163  STATUS,SUSPENDED  ACCESS,SECURE
VIEW ALIAS,TEST2  LINC-SYSTEM,RISTO  STATUS,UP  ACCESS,SECURE
(more views)
LANGUAGE  LINC,FRENCH  ISOLANGUAGE,FR  ISOCOUNTRY,CA
LANGUAGE  LINC,ENGLISH  ISOLANGUAGE,EN  ISOCOUNTRY,US
```

In this example, there are two servers, using CMS 1100 TSAM; both servers can schedule transactions in application group 8. The PID allocation ranges are verified across all servers.

```
SERVER   NAME,ALSRV2   PORT,4454  CPCOMM,0205222
ALSRV2  TSU,MEFIST  TSU-PWD,GLOVES   APP,9 APP,16
PID-INFO  PID-SERVER,ALSRV2  APP,9  START-PID,2010 NUM-PIDS,4095 LINC-STATUS,UP
PID-INFO  PID-SERVER,ALSRV2  APP,16  START-PID,2000 NUM-PIDS,10 LINC-STATUS,UP
VIEW ALIAS,BBI LINC-SYSTEM,BBIW3 STATUS,UP ACCESS,OPEN
```

In this example, there is one server, using CPComm. Note that this BDI reflects CPComm and not TSAM. It is possible to have one server assigned to CMS1100 and the other to CPComm with different port numbers assigned to each.

The following tables describe the configuration statements for servers.

Notes:

- *New server entries can be added at any time.*
- *The tags marked with an asterisk(*) in the tables can be changed dynamically through a reconfiguration.*
- *If your System uses TIP Session Control, the TIP Session Control settings will override the VIEW setting and the number of PIDs configured for the Remote Access Server. See "Using TIP Session Control" on page 3-40.*
- *Remote Access Server uses CPCOMM when no CPCOMM or TSAM-BDI entry is defined in the CONFIG element.*
- *You must use the same IP address that you used for CPCOMM or CMS, depending on where you configured the TSAM PROCESS statement.*

SERVER

Tag	Default	Description
NAME	None	The name of this server. The maximum length is 12 characters.
PORT	2449	Optional entry. The port number (1-65,535) on which this server should listen.
TSAM-BDI or CPCOMM	0204520	Optional entry. Contains the BDI of the TSAM or CPComm subsystem to use. It is only necessary to include the TSAM BDI if your site has multiple CMS 1100 products installed. If neither entry is included, CPCOMM is used.

Server-name

Tag	Default	Description
TSU	None	The name of the TSU as configured in CMS 1100 or CPComm. The maximum length is 6 ASCII characters.
TSU-PWD	None	The password associated with this TSU as configured in CMS 1100 or CPComm. The maximum length is 32 ASCII characters.
APP	None	The application group numbers in which the server can schedule transactions. Enter a value in the range 1 to 16.

PID-INFO

Tag	Default	Description
PID-SERVER	None	Matches a NAME tag. The maximum length is 12 characters.
APP	None	The application group for which a range of PIDs is to be allocated for this server. Enter a value with in the range 1 to 16.
START-PID	None	The first pseudo PID that this server can use when scheduling a transaction in a particular application group. The PID ranges cannot conflict with PIDs defined in the CMS 1100 or CPComm config or any other product that registers itself as a CMS with EXEC.
*NUM-PIDS (increase only)	None	The maximum number of pseudo PIDs that can be used in this application group for this server.

Tag	Default	Description
*LINC-STATUS	None	Indicates whether this server is allowed to schedule transactions in this application group. UP allows new transactions to be scheduled. DOWN prevents any messages from being scheduled. SUSPENDED inhibits new transactions from being scheduled.

Note: *If your System uses TIP Session Control, it will override the NUM-PIDS setting with the system application group setting for the number of concurrent sessions.*

VIEW

Tag	Default	Description
ALIAS	None	The name by which the client knows the System.
*LINC-SYSTEM	None	Name of the System Specification. The maximum length is 10 characters.
*STATUS	None	Indicates whether the server is allowed to schedule transactions for this VIEW. UP allows new transactions to be scheduled. DOWN prevents any messages from being scheduled. SUSPENDED inhibits new transactions from being scheduled.
*ACCESS	None	Indicates which of the User Hook security validation routines are to be called. OPEN will cause the server to invoke the routine RLOGPB. SECURE will cause the server to invoke the routines RLOGSC and RLOGDF.
*DEFAULT-USERID	RATL	Optional entry. Allows a configuration to optionally specify an LSM user id to be used for Component Enabler clients.
*DEFAULT-PWD	RATL	Optional entry. Sets the corresponding LSM password for the LSM user id.

Notes:

- *View entries can be added or removed. Changes to View entries become activated after LRU validation processing.*
- *The DEFAULT-USERID and DEFAULT-PWD values are not case sensitive. IF LSM is turned on for the target System, ensure that LSM has these value pairs configured as upper case.*
- *If your System uses TIP Session Control, it will override the ACCESS setting, making all sessions SECURE, without invoking the routines RLOGSC and RLOGDF.*

LANGUAGE

Tag	Default	Description
LINC	None	Optional entry. Language name as defined in Enterprise Application Runtime.
ISOLANGUAGE	None	Two-letter ISO standard language identifier, as defined by ISO 639.
ISOCOUNTRY	None	Two-letter ISO standard country identifier, as defined by ISO 3166.

Note: *The ISO Language and Country code standards are available on the official ISO website, http://www.iso.ch/.*

Processing Configuration Elements

Use the LRU utility to process this configuration element. This utility validates the contents of the configuration element and writes an output server readable element. To run the LRU utility, type:

```
@<env>*UTIL$.LRU <function>,<optional parameter>
```

Specify a <function> of RATLREPORT to validate the contents of the configuration file for the particular server. RATLREPORT performs an integrity edit and reports violations of the syntax rules. It produces a report based on the processing of the configuration. This command produces a server readable configuration but does not write it to the **LINC*RATL-SERVERS** configuration element.

Specify a <function> of RATLPROCESS to produce the server configuration file. For each System (TAG type LINC-SYSTEM) defined in the configuration, the associated SYS$LIB file for that System is inspected to determine the application group and transaction code used for that System. This allows the transaction codes and application groups of Systems to be changed without changing the configuration file; however, the configuration file will need to be reprocessed for changes to take effect.

Specify the name of the server you wish to work with in the <optional parameter> field. If this field is left blank, you will be prompted to supply a value.

Option B (Optional)

The execute option (B) is optional and is supported for both RATLPROCESS and RATLREPORT. Use the following to breakpoint your LRU output:

```
@<env>*UTIL$ LRU,B<function>,<optional parameter>
```

Where <function> is RATLPROCESS or RATLREPORT and <optional parameter> is the name of the server.

Administering Your Remote Access Server

An operator can start a runstream for the server manually, or a user with appropriate privileges can start one from a demand session. A sample runstream is installed in LINC*RATL-SERVERS.RATL-SERVER or <runtime>*UTIL$.RATL-SERVER when Runtime is installed. The following example shows a runstream to start the server:

```
@RUN,A/W RUNID,ACCNT,EA Runtime Name
@ASG,T REMP.,F/0/TRK/300
@ASG,A UTIL$.
@COPY,A UTIL$.RATLSRV,TEMP.
@FREE UTIL$.
@TEMP.RATLSRV
@FREE,D DIAG$
@FIN
```

Once the server runstream has been started, the server registers its generated Runid as a KEYIN$ service. You can use the KEYIN$ service to control the server activity. The following commands are recognized by the KEYIN$ service:

Command	Function
STATUS	This command lists the status of the server.
TRACE [ON/OFF]	This command enables or disables the ability to trace a connection if the client has requested tracing. This client tracing request can be specified on the protocol connect request to the server. This command without a subcommand of ON or OFF will list the current trace setting. The server can be started with tracing ON with a T option on the server call. The trace file is written to <server>*RATLLOG$, where <server> is the name of the Remote Access Server. For example, if the server is called ALSRV2, the trace file is ALSRV2*RATLLOG$.
DEBUG [ON/OFF]	This command enables or disables the ability of the server to write debugging information to the log file. This command without a subcommand of ON or OFF will list the current debug setting. The server can be started with debug ON with a D option on the server call.
RECON server-name	This command causes the server to re-read the configuration file and effect any changes. See the table of SSG vales that can be dynamically updated while the server is running.
STOP server-name	This command will cause the server to terminate provided there are no active connections.
TERM server-name	This command will cause the server to terminate unconditionally.
STATUS VIEW view-name	This command will list the view information of a specific View. Note that you cannot list the status of all Views as the amount of information displayed to the console could be overwhelming. Its primary use is after a reconfiguration takes place. The reconfiguration process will list those Views which have been updated.
SWAP	This command directs the server to close and free the log file and cycle up to a new one.

Security

If the System that your server is connecting with has password aging for its users, then the Remote Access server password will be assigned a password time limit. After this time the password will be invalid and a new password must be supplied.

If your System has TIP Session Control installed, its password aging can be set in SIMAN.

Message Queuing

Message Queuing allows multiple concurrent users of Component Enabler to access an Enterprise Application Environment System, by making more efficient use of system resources, and reducing the overheads associated with establishing and maintaining individual connections for each user session. This is an alternative to the existing method of connecting to the Runtime System using individual TCP/IP connections.

See your *Enterprise Application Component Enabler Developer's Guide* for further details of using message queuing for Component Enabler Scalability.

Installation Requirements

In order to use Message Queuing the following software is required:

- Enterprise Application Runtime Release 3.2 or later
- Remote Access Server
- Communication Application Program Interface (COMAPI) for OS 2200

  See your *Communications Application Program Interface (COMAPI) User's Guide* for installation instructions.
- FalconMQ client for OS 2200

  See your ClearPath IX Series *FalconMQ Client for OS 2200 Installation and Programming Guide* for installation instructions.
- Microsoft Message Queuing Server (MSMQ) on the client

Configuring the Server for Message Queuing

To enable the server to use Message Queuing, you must enter the following configuration statements at the beginning of the LINC*RATL-SERVERS.CONFIG element. See "Configuring the Remote Access Server on OS 2200" on page 3-32 for details.

MQSERVER

Tag	Default	Description
IP	None	The IP address of the FalconMQ server. The address must be enclosed in sets of two single quotation marks (″ ″).

Tag	Default	Description
DOMAIN	None	The domain name of the FalconMQ server.
PORT	None	The port number for the FalconMQ server.
FOR	None	The name of the Remote Access server.

MQUSER

Tag	Default	Description
NAME	None	The user name. This parameter is case sensitive.
PWD	None	The password associated with the user name. This parameter is case sensitive.
REQ	None	The name of the request queue. The name must be enclosed in sets of two single quotation marks (‘’ ‘’).

For example,

```
MQSERVER IP,''192.147.253.7'' DOMAIN,MQ PORT,1234 ;
                                          FOR,ALSVR1
MQUSER NAME,unisys PWD,acus REQ,''.\PRIVATE$\2200mqtest''
```

Once you have entered the statements:

1. Stop the Remote Access Server, if it is running.
2. Run the LRU utility by typing;

   ```
   @<env>*UTIL$.LRU RATLPROCESS,<Remote Access server name>
   ```

Administering Message Queuing

The following table describes the KEYIN$ commands that can be used to administer Message Queuing on the server. See “Administering Your Remote Access Server” on page 3-5 for further details.

Command	Function
MQSTART	This command attempts to start the Message Queuing service.
MQSTOP	This command stops the Message Queuing service.
MQSTATUS	This command displays the Message Queuing status, its connections, and relevant information.

Using TIP Session Control

TIP Session Control is a security application for ClearPath OS 2200, which enforces the input of a valid userid and password, and, optionally, account and project ids and a new password on password expiry. See your EXEC documentation for installation instructions.

If TIP Session Control is installed at your site, the Remote Access Server automatically enforces secure session access to Enterprise Application Systems. Even if OPEN session access is configured, all users must provide login details.

The following settings can be configured in SIMAN, and will override any corresponding Remote Access Server configuration settings:

- Number of concurrent sessions. This is a system application group setting, and applies to all applications on the host, including Remote Access Server.

  You should ensure that the NUM-PIDS setting in your LINC*RATL-SERVERS.CONFIG element is not more than the number of concurrent sessions for TIP Session Control. If you configure more Remote Access Server PIDs than concurrent sessions allowed on the host, any users connecting to Remote Access Server after the number of concurrent sessions has been reached will be denied access.

  For example, the number of concurrent sessions could be set in your TSC configuration as follows:

  ```
  . *********** APPLICATION GROUP 3 udssrc **********
  STEPCONTROL  3 QNBR IS 3000
  STEPCONTROL  3 SAVEFILE IS EXEC,SYS$,APP3SAVE,,,D
  STEPCONTROL  3 NUMBER OF CONCURRENT SESSIONS IS 1000
  STEPCONTROL  3 SESSIONFILE CONTAINER IS TIP$,TSC3
  STEPCONTROL  3 TIP SESSIONFILE IS TSC3,300
  .
  .
  .
  . *********** APPLICATION GROUP REQUIRED FOR APPNIN *********
  STEPCONTROL 9 APPLICATION NAME IS APPNIN
  STEPCONTROL 9 QNBR IS 1000
  STEPCONTROL 9 SAVEFILE IS EXEC,SYS$,APP9SAVE,,,D
  STEPCONTROL 9 NUMBER OF CONCURRENT SESSIONS IS 1500
  STEPCONTROL 9 SESSIONFILE CONTAINER IS TIP$,TSC9
  STEPCONTROL 9 TIP SESSIONFILE IS TSC9,309
  ```

  The NUM-PIDS could be set in your Remote Access Server configuration as follows:

  ```
  SERVER   NAME,RTLTST   PORT,4454  TSAM-BDI,0204520
  RTLTST  TSU,MEFIST  TSU-PWD,GLOVES   APP,9
  PID-INFO  PID-SERVER,RTLTST  APP,9  START-PID,2010 NUM-PIDS,100 LINC-STATUS,UP
  VIEW ALIAS,ALB31B LINC-SYSTEM,ALB31B STATUS,UP ACCESS,SECURE
  VIEW ALIAS,ALB31B LINC-SYSTEM,ALB32  STATUS,UP ACCESS,OPEN
  ```

- Session time out for each userid. Without TIP Session Control, sessions can only be closed by a log-off request or fatal error.
- Password expiry. If a password has expired, the user will be prompted for a new password.

- Prompt for account number. If this setting is enabled, the user will be prompted to enter their account number.
- Prompt for project id. If this setting is enabled, the user will be prompted to enter their project id.

See your SIMAN documentation for details of configuring TIP Session Control.

The following table gives details of login prompts.

Prompt	Details
Userid	Maximum 12 alphanumeric characters Required prompt
Password	Maximum 6 alphanumeric characters Required prompt
Account number	Maximum 12 alphanumeric characters Optional prompt
Project id	Maximum 12 alphanumeric characters Optional prompt

Using Hooks

For Systems without TIP Session Control, there are three user hook routines supplied with the server that allow a site to tailor its security. The server allows either OPEN or SECURE access to the System.

Open Access

OPEN access allows clients to connect to the application without providing login authentication details. In this case, a hook routine called RLOGPB is called by the server whenever a client connects to a View that has OPEN access specified. The RLOGPB hook can perform whatever validation the site requires and returns an indicator to signify whether the authentication was successful or not.

If Security Module (LSM) security is turned on for the System, DEFAULT-USERID and DEFAULT-PWD values must be specified in the configuration file and defined in LSM. The default RLOGPB is a routine that always returns a successful validation status.

Secure Access

For SECURE access, the server requires two supporting hook routines. These are RLOGDF and RLOGSC.

RLOGDF returns the names of authentication fields together with their attributes. For instance, the default routine provided with the server returns two field names called *User Id* and *Password*. The attribute of the *User Id* field is that its value can have 12 characters. The attributes of the *Password* field are that its value can have 12 characters and when the user enters the value into this field the characters are not to be echoed. Up to five items of authentication can be defined.

The LSM userid and password are case sensitive, but the operating system is not. Therefore the user should ensure that the userid and password entered in the client are the same as the LSM userid and password.

RLOGSC is responsible for validating the values that are passed from the client. In the default RLOGSC provided with the server, the userid and password values are validated by checking that these values are defined in SIMAN for Demand access.

Since the default routines simply pass the values to LSM, it is important that the same userids and passwords are defined in Demand and LSM.

If you wish to have the access validated purely by LSM, you need to change the default RLOGSC routine to simply return without performing the Demand mode userid and password validation.

Note: *As Remote Access Server cannot perform dynamic linking at runtime, user hook routines must be fully linked to resolve any external references (that is, entry points must be defined even if hooks are not being used). This also applies to Systems using Tip Session Control.*

Configuring the Remote Access Server for Enterprise Application Developer

When combined with Component Enabler, the Remote Access server in conjunction with Developer Test creates a complete development and testing environment. Alternatively, the same functionality can be achieved with Graphical Interface Workbench, the Remote Access Server, and Developer Test.

See the *Unisys Enterprise Application Component Enabler User Guide* for details of using Component Enabler. See the *Unisys Enterprise Application Developer User Guide* for details of using Developer Test.

The Remote Access server configuration information is stored in the ActiveLINC.INI file. You should not modify the ActiveLINC.INI file directly. The dialog-driven **Developer Test Component Enabler Configurator** is provided to allow you to safely add the data you need to configure the Remote Access server.

Before you can use the Remote Access server, you need to configure it and create Views using the Developer Test Component Enabler Configurator. You can access this utility in the following ways:

- Through the the command line, by entering ALCONFIG.EXE.
- On the Windows Start menu, by selecting Programs > Enterprise Application Developer > Developer Test Remote Server > Configurator.

Working with Views

You can perform the following actions using the buttons on the Developer Test Component Enabler Configurator.

Adding a View

1. Open the **Developer Test Component Enabler Configuration** utility.
2. Click the **Maintain Views button**. The **View Maintenance Utility** is displayed.
3. Click the **Add** button. The **View Details** dialog box is displayed.
4. Enter details in the following fields:

    - **View name**

      The name to identify the View.

    - **Business Segment name**

      This field is mandatory. Enter the name of the Business Segment to which the View belongs.

    - **LINC.INI path**

      You need to enter the path if the desired LINC.INI file is not in the same directory as the LINCRUN.EXE. For example, the default installation copies LINCRUN.EXE to the C:\Program Files\Enterprise Application Developer directory, and LINC.INI is installed to the C:\Program Files\Enterprise Application Developer\Work directory. In this case you need to enter the appropriate LINC.INI directory path.

    - **Repository path**

      You need to enter the path if the desired repository is not the one specified in the LINC.INI file.

    - **Developer Working Directory**

      You need to enter the path if the desired working directory is not the one specified in the LINC.INI file.

    - **Host Data Access**

      This group of fields enables the View to access a deployed test database on a host. It is important that the System and Option Set names you enter are correct as no validation is carried out until runtime. See your *Enterprise Application Developer User Guide* for details of testing databases on a host.

      Access Host Database

Check this check box to access the deployed database for the specified system.

> **Caution**
>
> This facility is for testing purposes only. Do not use it with your deployed production databases. Applying untested code to a production database could destroy the integrity of the database, risk loss of data, and risk database corruption.

Enable Remote CALL statements

Check this check box to access the host when processing CALL; statements.

System

Enter the name of the system to be accessed on the host, as defined in Developer.

Option Set

Enter the name of the Generate Set defined for the specified system. The Enable Host Database Access and/or Enable Remote CALL Statements options must be set on the Developer Test page of the Business Segment options dialog box.

5. Use the radio buttons in the **Logging** field to select your logging options.

   You can set logging on or off, or you can set logging at the client's request.

6. Select a radio button in the **Host** type field.

   Select the host type corresponding to the Target Host set in the Business Segment options, either Clearpath MCP or Other.

7. Click the **Login** button to open the **Login Screen Details** dialog box. You can customize the following details:

   - Username label
   - Username Greeting
   - Domain label
   - Domain Greeting
   - Password label
   - Password Greeting

   This information is displayed when you connect to the ComponentEnablerViewer. You can also access this information when you use the Component Enabler API and display them wherever you want.

8. Click the **OK** button to return to the View Details dialog box.

9. Click the **Security** button to add security information. The **Security Values** dialog box is displayed.

   Use the following fields to set access permissions for the View:

   - **IP Addresses**

     Define a list of IP addresses, including IP ranges (123.123.12.*). Each entry should be separated by either a space or a semicolon (;).

     Select the **Permitted** radio button to specify that only users from the specified addresses are allowed to access the View.

     Select the **Forbidden** radio button to specify that users from the specified addresses are automatically denied access to the View.

   - **Users**

     Define a list of user ids. List the user ids in the format domain1\userid1; domain2\userid2; .... Where the user is defined on the local machine use .\userid. Each entry should be separated by either a space or a semicolon (;).

     Select the **Permitted** radio button to specify users in the list are allowed to access the View.

     Select the **Forbidden** radio button to specify users in the list are automatically denied access to the View.

     Select the **Anonymous** radio button to specify anonymous user support for the View. Specify an anonymous login that will initiate a connection to Developer Test using the user id, ALPublic, and the password and domain name defined for ALPublic.

10. Click the **OK** button to return to the View Details dialog box.
11. Click the **OK** button.

Modifying an Existing View

1. Open the **Developer Test Component Enabler Configuration** utility.
2. Click the **Maintain Views** button. The **View Maintenance Utility** is displayed.
3. Select an existing View from the list.
4. Click the **Edit** button. The **View Details** dialog box is displayed
5. Change the desired fields. Click the **OK** button.

Note: *Adding and changing Views takes effect immediately.*

Deleting a View

1. Open the **Developer Test Component Enabler Configuration** utility.
2. Click the **Maintain Views** button. The **View Maintenance Utility** is displayed.
3. Select an existing View from the list.
4. Click the **Remove** button.

   A prompt dialog box will ask you to confirm that you want to delete the View. Click **Yes** to complete the deletion. Click **No** to cancel the process.

Using the Language Mapping Utility

1. Open the **Developer Test Component Enabler Configuration** utility.
2. Click the **Language Mappings** button. The **Language Mappings** dialog box is displayed. It contains the following fields:
    - The **Host Languages** list box contains all the languages for all the defined Views.
    - The **ISO Languages** list box contains a list of ISO-recognized languages.
    - The **ISO Countries** list box contains a list of ISO-recognized countries.
3. Select a combination of ISO Country and ISO Language to uniquely identify the language dialect.
4. Select the Host Language to which you wish to map the selected dialect, then click the **Map Language** button.
5. Click the OK button.

Note: *In the Developer environment, the **Host Languages** list is initially empty. The list is not populated until a connection is made to Developer Test. The list is updated each time a connection is made.*

Enabling Anonymous Login

1. Create the user ALPublic.
2. Open the **Developer Test Component Enabler Configuration** utility.
3. Click the **Public user details** button. The **Login details for user ALPublic** dialog box is displayed.
4. Enter the password and domain name for use with this user id. Click the OK button.
5. Define the ALPublic user id to the Windows User Manager.
6. You must have selected the **Anonymous** radio button in the **Security Values** dialog box, to specify anonymous user support for the View.

Changing the Service Settings

1. Open the **Developer Test Component Enabler Configuration** utility.
2. Click the **Service Settings** button. The **Service Settings** dialog box is displayed.
3. Use the port fields to change the TCP/IP connection ports. Ensure that the values you use are not used for any other TCP/IP service.
    - The **Listener port** defaults to 4323.

      This is the TCP/IP port on which the Component Enabler client is connected.
    - The **Developer Test port** defaults to 4324.

      This TCP/IP port is used for interprocess communication between the Remote Access server and Developer Test.

4. Check the **Enforce Security** check box to enforce Windows security when a user connects to the Remote Access server. To use this feature, the user running the server must have the “Act as part of the operating system” right granted. To do this:

   - Access the **Windows User Manager**, Start > Programs > Administrative Tools (Common) > User Manager.
   - Select the user that is to run the Remote Access server.
   - Select the **User Rights** command on the **Policies** menu.
   - Check the **Show Advanced User Rights** check box.
   - Select **Act as part of the operating system** from the **Right** drop-down list.
   - Click the **Add** button, then click the **OK** button.

   By default this check box is unchecked. Any entries from clients will be accepted. It is not necessary for the usercode enter to be a valid Windows usercode.

5. You cannot change the number in the **Max Connections** field. Click the **OK** button.

Administering Your Remote Access Server

Starting and Stopping

Each time a Remote Access server connection is established a Developer Test session is initiated. Sessions can be started from the same machine as the installation of Developer Test or from a separate machine running the Component Enabler Client. In this release the number of sessions that can be initiated is limited to one.

Logging

You can set your logging options for a View. These options are for automatic logging on or off, or for logging at the client's request. If the last option is specified, the client can initiate tracing in its connection request.

All errors are written to the errors.log file in the %LINCIIDIR%\log directory. When logging is turned on, additional tracing messages are logged to the ratltrace.log in the same directory.

Using the Remote Access Server with Graphical Interface Workbench

Creating a Graphical Interface Workbench Script

In order to create a Remote Access server connection from the Enterprise Application Workbench or from Web Enabler to a Runtime System, you require a script.

Graphical Interface Workbench scripts are created using the master script templates provided. To use the Remote Access server, you must create a script using the "LINC 16 RATL" master script template. The same script template is used to connect to Windows operating systems hosts, UNIX operating system hosts, Developer Test hosts (when connecting from Graphical Interface Workbench), MCP based hosts, and OS 2200 based hosts. Scripts are stored as objects in the Graphical Interface Workbench Repository.

A Remote Access server script is the recommended way of connecting to a Runtime System, rather than using a Winsock script or a script that uses an INFOConnect path. One advantage of using the Remote Access server is that it does not require INFOConnect Connectivity Services. This means that INFOConnect does not need to be installed on each workstation along with Graphical Interface Workbench. In addition, any Big Buffer Ispecs larger than 1987 bytes in your System are not supported by INFOConnect, due to the INFOConnect buffer size limitations.

The Remote Access server script contains the information required to establish the connection to the host, log onto Enterprise Application Runtime, and connect you to the View specified in the script. Graphical Interface Workbench uses the Windows Socket Library (Winsock.dll) to establish the connection to the host. Remote Access server defines a packet protocol that is sent across a socket connection.

Creating a Script

1. Start Graphical Interface Workbench and log on as an administrator.
2. From the User menu, select **Script Administration** and then select **Create Script**.

   The User Script dialog box is displayed.
3. In the **User Script Name** field enter a name for the script, for example, RATL-SAMPVIEW-LCOM. This script creates a connection to the View named SAMPVIEW, and LCOM is the name of the computer where the Runtime System is running.
4. Click the **OK** button.

   The Selection dialog box is displayed.
5. From the list select the “LINC 16 RATL” master script template.
6. Click the **Select** button.

   The main script dialog box is displayed.
7. Click the **OK** button to complete the script.

Running a Script

If you are using the Remote Access server to connect to a System from the Enterprise Application Workbench, complete the following procedure:

1. Start Graphical Interface Workbench.

2. Start Enterprise Application Workbench from the main Graphical Interface Workbench window.
3. Select the **Open Session** command on the File menu.
4. Select the name of the script in the dialog box, and click the **Select** button to run the script.

If you are using the Remote Access server to connect to a Runtime System using Web Enabler, use the script name to connect to the System (specified in the URL that connects you to the System from your Web browser).

Summary

After completing this section you can:

- Perform the actions necessary to configure the server on a specific platform.
- Understand the basic administration tasks for each platform.
- Create a Remote Access server script to connect Graphical Interface Workbench to all platforms.

Appendix A
Related Product Information

The following publications contain information relevant to the use of Component Enabler and PowerClient as well as the definition and operation of Systems on specific platforms. These publications are reference sources for users who have completed training courses. See your local Unisys representative for information on available training courses.

These documents are published by Unisys Corporation and are available on the Internet at http://www.support.unisys.com. Order hardcopy documents online through the Unisys Book Store at http://www.app1.unisys.com/bookstore.

Unisys Enterprise Application Component Enabler Developer's Guide

This document provides an overview of how to install, generate, and deploy Component Enabler applications.

PowerClient Installation and Configuration Guide

This guide describes how to install Graphical Interface Workbench software and configure the workstations and the Repository.

PowerClient Development Studio User's Guide

This guide provides information on how to use the development tools and utilities in the Development Studio environment. These tools and utilities allow a user to convert host legacy screens to client builder graphical user interface (GUI) applications that can communicate with a Enterprise Application, Business Information Server, or 3GL application through the Graphical Interface workbenches.

Unisys Enterprise Application Getting Started with Runtime for the Windows 2000 Operating System

This document describes the installation, configuration, and initial use of Runtime for Windows Operating Systems.

Unisys Enterprise Application Runtime for the Windows 2000 Operating System Administration Guide

This document provides an overview of the Runtime for Windows Operating Systems environment and details on the generation and administration processes.

Unisys Enterprise Application Runtime for ClearPath MCP Installation and Configuration Guide

This document contains procedures for installing and configuring released software on MCP based hosts.

Unisys Enterprise Application Runtime for ClearPath MCP Administration Guide

This document describes the generation and operation of Systems and Reports, and the general administration of Systems for MCP based hosts.

Unisys Enterprise Application Runtime for ClearPath OS 2200 Installation and Configuration Guide

This document describes the procedures for installing and configuring released software on OS 2200 based hosts.

Unisys Enterprise Application Runtime for ClearPath OS 2200 Administration Guide

This document describes the generation and operation of Systems and Reports, and the general administration of Systems for OS 2200 based hosts.

Unisys Enterprise Application Runtime for the UNIX Operating System Administration Guide

This document describes the generation and operation of Systems and Reports, and the general administration of Systems for UNIX based hosts.

Unisys Enterprise Application Runtime for the UNIX Operating System Installation and Configuration Guide

This document describes the procedures for installing and configuring released software on UNIX based hosts.

Unisys Enterprise Application Developer Installation and Configuration Guide

This document provides information on the installation and configuration of Developer.

Unisys Enterprise Application Developer User Guide

This document provides information on how to perform administration tasks for Developer.

Unisys Enterprise Application Builder Guide

This document provides information on using Developer Systems and Builder to define a System for generation directly to a target Enterprise Application host.

Builder is licensed separately from the basic Developer components. If your site has not purchased Builder then you do not need this document.

Glossary

A

access class

For Ad Hoc Inquiry, a particular type of user access to queries, defined by a letter (A through Z) or an asterisk (*, for unrestricted access). Also called *query class*.

Action line

A field that appears on most screens in mainframe Enterprise Application Systems, enabling fast-track navigation to required functions. Also referred to as the **Action** field.

Activity

A group of Objects which together perform a business function.

Active Server Page

A type of HTML page, sometimes abbreviated to ASP, which can create and run script commands and Active X componnets to create interactive Web pages or Web based applications.

Ad Hoc Inquiry

An MCP based mainframe facility used for making inquiries on an Enterprise Application Database.

Administration Client

A tool for administering Enterprise Application user systems and their associated databases. The Administration Client can be used on a remote workstation or on the runtime server. The Administration Client communicates with the Administration Services using Remote Procedure Call (RPC) to perform the requested operations.

Administration group

A group of user accounts within a Windows domain for people who have privileges to administer Systems, for example, to add a new database to an Enterprise Application environment. The Administrator is a separate role from the Windows Administrator, although the two roles may be performed by the same person. Most administration functions are performed from the Administration Client interface.

Administration Guide

The manual that contains instructions for administering and operating Enterprise Application Systems on a particular host type.

Administration Services

The set of Windows services on the runtime server that communicates with Administration Clients using RPC. These services perform various administrative functions on behalf of the Administration Clients.

Administrator

Person responsible for administration of a computing environment.

archive

Copy files to a directory as protection against accidental loss, deletion, or damage.

Audit Archive File

An archive file of audit records.

Audit Domain

An Audit Domain defines the scope of audit activity for Developer. An Audit Domain can be a Model or a Business Segment.

auditing

Auditing records events that occur in the Developer and Developer Security environments, providing a history of changes made to the Developer Repository.

Audit Store

The Audit Store holds audit records. Because the function of auditing is to record operations that affect a Developer Repository, there is one Audit Store for each Repository.

Automaint Memo Component

A Memo Component with one or more Ordinates defined by a specified Profile (that is referred to as its *Automaint* Profile). *See also* Component, Standard Component and Automaint Profile.

Automaint Profile

The Profile of an Automaint Memo Component that provides automatic access to the records of that Component. *See* Automaint Memo Component, Profile.

Automatic Entry

(1) A facility that enables an Ispec to create a record in another Ispec. (2) A facility that enables an Ispec to modify an existing record in a Standard Component, Table Component, or Automaint Memo Component. Not applicable to Copy.From Ispecs. *See also* Automatic Entry buffer, external Automatic Entry, internal Automatic Entry, HUB.

Automatic Entry buffer

An area in memory used by the Automatic Entry facility. *See* Automatic Entry.

Automatic Lookup

A process where a Component record is automatically read into memory from the database.

Automatic Profile

The Profile created by Developer for a Component with an Ordinate. This Profile cannot be seen in the Model Directory, but can be viewed as part of a System in the Systems Directory.

automatic undo management

In Oracle, a method for rolling back changes to the database using information stored in an undo tablespace, instead of manual rollback using rollback segments.

B

Background Run

For OS 2200 Systems, a background run that controls certain functions; for example recovery, Report handling, and setting up of Common Banks. There is one Background Run for each Runtime.

Background window

The window behind the first window you see when Developer is initiated. The Background window provides access to the various functions of Developer.

backup

Make a copy of a database or subset of a database.

banner

A system-generated page that prints at the start of a Report and provides control information about that Report.

base year

The year upon which the DATE.CONVERT; command bases relative day numbers. Base year is defined using the Business Segment dialog box in Developer. It may be accessed through the System Data Item GLB.BASE. *See* relative day number.

Big Buffer Ispec

An Ispec capable of receiving input greater than the usual limit of 1920 bytes. *See also* Ispec.

binary

Object files that may be .int, .gnt, .exe, or .dll files. A runtime transfer using export or import transfers the runtime files, not the source COBOL files.

bind

The task of coupling object files with executable files for dynamically linked program code. Ensures that externally referenced procedures or subprograms are available to a main program.

breakpoint

A feature of the Developer Test Debugger that stops logic execution in response to a certain state of the System.

buildable LINCIIDIR

The Enterprise Application installation directory is defined by the environment variable LINCIIDIR. A LINCIIDIR that contains the object packages necessary to build executables using object packaging is a buildable LINCIIDIR. A buildable LINCIIDIR is necessary to use object packaging to maintain your Enterprise Application environment.

Builder

The software used to generate Specifications (Business Models) as applications. Generation is the process of generating a complete set of source files and then compiling and linking those files to create a set of executables. Through Builder, a Developer workstation can generate an application to a target host runtime environment.

Builder Client

The Builder Client runs on the workstation that initiates the generate process and controls the generation of the System according to its specified elements and options. It also includes the Generate Client/Server interface to the Builder Server.

Builder Server

The Builder Server runs on the host to which the Builder Client is connected. It compiles the Enterprise Application System generated by the client and provides the runtime environment for the application.

Bulk Options

A tab on the Business Segment Options dialog boxes in Builder that enables options for several Ispecs, Reports, Profiles, or Global Logics to be defined together.

Business Model

Repository containing details of Business Segments (Specifications) held by Developer.

Business Rules

Information entered as text which supports an Element.

Business Segment

Part of a Model that provides the definition for an Enterprise Application System in Developer.

C

Change control

A method for tracking changes to Specifications.

Change identifier

A alphanumeric value used to identify a single user or user terminal for change control purposes. Applies when the Change Identification option is selected for that Specification. *Contrast with* Patch identifier.

CHG

Object Changes (CHGs) are used to distribute new features, fixes, customer requests and newly validated versions of support software on the ClearPath OS 2200 platform. *See also* IC.

class

See access class, security class.

client

A Windows program, typically running on a workstation, that works cooperatively with one or more programs or services running on a server computer. In some cases, the client program may reside on the server computer, but uses the network interfaces to communicate with the server program or service. In Runtime for Windows Operating Systems, there are two distinct types of client:

1. An operational tool, such as the Administration Client, which deploys, manages, and maintains user systems.
2. An end-user client, such as Graphical Interface Workbench, that provides workstation users access to Enterprise Application user systems. This class of client includes user-written clients developed in languages such as C++ and Visual Basic.

Client Listener

This service listens for and accepts connection requests from the Administration Client.

client/server

A distributed architecture in which client workstations communicate with servers through a network. For instance, a client typically provides initial processing, data gathering functionality and the user interface. It then communicates the data and requests to a server for further processing.

Column name

For Ad Hoc Inquiry, an SQL term equivalent to Data Item.

compile

To create object files from source files.

Component

(1) A business resource such as a customer, product or vendor. (2) A store of static data about a business resource. Consists of a screen layout and associated logic (Pre-Screen logic, Pre-LINC logic, and Main logic). Together with Events, Components form the fundamental building blocks of a System. *See also* Ispec, Standard Component, Table Component, Memo Component, Automaint Memo Component.

Component Enabler

The product with which users can build their own GUI interfaces, or Views, to Systems. These Component Enabler applications use the Remote Access Server to communicate with Systems on the host environment. Component Enabler allows applications to use current Web technology and the Enterprise NT world. *See also* Remote Access to Enterprise Application.

Component Profile

For Ad Hoc Inquiry, a Profile over one Component. *Contrast with* Event Profile.

Component record

For Ad Hoc Inquiry, an individual database record for a Component. Equivalent in SQL terminology to a *row*.

COMSTP Program

A pre-compiled program that contains the necessary logic for routing of user transactions in a MCP based System.

COMUS

For OS 2200 based Systems, a product used to build Runtime before installation. *See* SOLAR.

Control tables

Tables created for the control of an environment within each Oracle database (SID) associated with the environment. Examples are PROCTAB and LSYS.

Copy.From Ispec

An Ispec with a screen that has a number of identical recurring lines. Transmission of the whole screen produces a separate record in the database for each completed line. *See also* Ispec.

Critical Point

A user-specified recovery point within a Report. In the event of a failure, recovery will restore the environment to the last successful Critical Point, then resume execution at the location of the Critical Point.

D

DAD

See Data Display.

Data Attribute

An attribute of a Data Item. For example, LENGTH, EDIT, BRIGHT.

Data Command

A command that identifies or creates a Data Item used in a screen layout, Report, or logic.

Data Dictionary

A collective name for the Local and Global data dictionaries. *See* Local Data Dictionary, Global Data Dictionary.

Data Display

Optional descriptive text for a Data Item, used in status messages in place of the name of the Data Item. Defined using the Data Display field on the Screen Data Item – Data Attributes dialog box.

Data Item

The variable used to enter, manipulate, store, and display data. For example, Ordinates, Setup Data Items.

Database ID

The two-character identifier for either an Oracle or SQL Server database. This identifier was previously referred as the Oracle SID. *See* Oracle SID.

Database Management Utility (DMU)

A utility that enables you to maintain your Enterprise Application Database. Use DMU to perform such tasks as changing the amount of disk space allocated to database structures, initiate a full or partial database reorganizations (for MCP based Systems), and creating Oracle SIDs (for Systems based on the UNIX operating system).

debug settings

A set of breakpoint, watch, and debug options that can be saved to a file and loaded into a Developer Test Debugger session as needed.

DEPCON

The Unisys DEPCON software application is a comprehensive print-management and file distribution solution for mixed-platform networks. Coupling Enterprise Application Environment with DEPCON increases the design possibilities and flexibility of Reports. See the *DEPCON Software Configuration and Operations Guide* and "Using DEPCON Reports" in the *Enterprise Application Developer User Guide*.

deploy

To implement a set of Enterprise Application executables and a database on a host machine. The result of deployment is the creation of an application on the host.

Deployment Server

The Windows server on which an application is installed and executed. Does not have Micro Focus COBOL installed.

Deployment Services

Windows programs installed with Runtime for Windows Operating Systems to perform specific functions, such as listening for and accepting connection requests on behalf of an environment. A collective term for the Administration and Generation services.

Design Audit

Analyzes a Business Segment, Functional Area, or Activity, and lists any items that may indicate poor design that will impact performance in a generated System.

Developer

Workstation based development tool for creating and maintaining Business Models (Specifications). From Developer, Business Models are transferred to host based Builder for generation and deployment, or generated directly to a target host. Formerly known as LDA III or LINC Development.

Developer Console

Developer Console provides a command line interface to Developer and Developer Version Control.

This facility allows Developer to run in batch mode, providing the ability to automate time-consuming activities such as loading model files and populating the Developer Repository from the Version Control Bank.

For routine processes, a Windows scheduler (for example, WinAT) can be used to start Developer Console automatically.

Developer Test

A workstation testing environment for Developer which includes a logic Debugger. Formerly known as LINC Development Runtime.

Development Environment

A collective term for the tools used to design, develop, and generate Enterprise Application Systems. Includes Developer, Builder, and utilities.

Development System

See Developer and Builder.

Differences Report

An XML report of the differences found when two files, objects, or revisions, or two versions of a specification are compared. This report can be viewed and printed from an Internet browser, or manipulated as an XML file.

Direct Component

A Component type that uses a key for direct access to the dataset, saving on the number of Input/Output operations. It has a single numeric Ordinate with a maximum of 8 digits. *See* Component.

Direct Report

A Report that uses the Report Output Control System (ROC), and which sends output directly to an output device and not to the ROC database. *See also* Report Output Control System. *Contrast with* Standard Report.

Display item

A string of descriptive information displayed on an Ispec screen layout or Report Frame.

DLL

Dynamic Link Library, contains one or more functions that are compiled, linked, and stored separately from the processes using them.

DMU

See Database Management Utility.

E

Editor

An Enterprise Application Environment facility for specifying logic for Ispec and Report processing.

Element

Collective term for the individual parts of the Business Model. An Element may be any of the following: Activity, Functional Area, Language, Object, or Wildcard.

Enterprise Application Database

The database created in the generation of an Enterprise Application System. Used for storing data.

Environment

The operating environment and supporting services for applications on a Windows server. Each environment is created by a separate installation of Runtime for Windows Operating Systems. There may be multiple environments on a single Windows server. Each environment includes one or more databases. Each database may contain zero or more user systems.

Environment database

The database associated with an environment that contains control tables. These tables control the environment for all the user systems in the environment. Synonymous with control tables.

Event

(1) An activity performed by an organization, for example a sale, purchase, or payment. (2) A store of data about an activity performed. An Event consists of a screen layout and associated logic (Pre-Screen logic, Pre-LINC logic, and Main logic). Together with Components, Events form the fundamental building blocks of a System. *See also* Ispec.

Event file

For Ad Hoc Inquiry, the file containing the Event records.

Event Profile

For Ad Hoc Inquiry, a Profile over one or more Events. *Contrast with* Component Profile.

Event record

For Ad Hoc Inquiry, an individual database record for an Event. Equivalent in SQL terminology to a *row*.

Executables

Executables are one of two methods of distributing applications to customers. They are the compiled set of application files which are ready to install and run. Contrast this with *Objects*.

EXE file

A Windows native executable file.

export

To copy an Enterprise Application System to a file for transfer to another environment.

Extended Language Message System (ELMS)

An OS 2200 based facility that provides translatable versions of messages used by software.

extent

A contiguous block of disk space assigned by Oracle. Extents are used in data segments (table data storage), index segments, rollback segments, and temporary segments. An extent is a unit of database storage space allocation. When an extent is used up, Oracle allocates a new one.

External Automatic Entry

An Automatic Entry to or from an external source, such as another System. *See* Automatic Entry, HUB. *Contrast with* internal Automatic Entry.

Extract file

A non-database (text) file created or read by a Report.

F

Filegroup

A discrete part of an SQL Server database. Filegroups in the same database can be stored on different physical devices. *See* tablespace.

Fireup Ispec

The Ispec that is displayed on your terminal when you sign on to an Enterprise Application System.

Forms Translation Utility (FTU)

A utility that enables you to modify screen layouts, Teach screens, Data Displays, and translatable Global Setup Data Items of your System without regeneration. FTU can also be used to modify Enterprise Application Software.

Frame

A number of Report lines containing display text and Data Items that is output as a single unit when the Frame is invoked by Report logic.

Function Point Analysis

A technique for estimating the size of an existing or future information system, using a unit of measurement called a Function Point.

Functional Area

(1) A group of Activities. (2) A concept used to group Activities for the purpose of loading into Builder.

G

garbage collection

An Administration Client function that aids the compacting of Oracle database tables.

Generalized Interface (GLI)

A facility that enables a external program to initiate transactions into an Enterprise Application user system.

generate

Perform the tasks necessary to create an Enterprise Application System on a host, including:

- Importing Enterprise Application user system files from the development environment or another server
- Deploying a runtime System and a database

Generate Group

A specified group of Reports that can be generated together.

Generate Set

A named set of values for generating a particular System from a Specification; for example, for a particular host type.

Generate Threads

A number of parallel generate tasks that can be initiated on the workstation.

GLI

See Generalized Interface.

Global Data Dictionary

(1) A data dictionary that controls the use of Data Items over all Business Segments. (2) Part of GLOBAL Specification. (3) *See also* Data Dictionary, Local Data Dictionary.

Global Logic

Part of a Specification that can be reused in more than one part of your Specification or Business Segment. Inserted into Ispecs, Reports, or Global Logics by using the INSERT; logic command. *See* Insertable Global Logic, Performable Global Logic.

Global Setup Data Block

An internal logic block used to group Global Setup Data Items. Grouping Global Setup Data Items for example, by Ispec would mean that when you recompiled that Ispec for generate purposes, only those Global Setup Data Items grouped in that Block would be recompiled, rather than all of them. Global Setup Data Items can only be defined within Global Setup Data Blocks.

Global Setup Data Item

A type of Setup Data Item that can be used by any Ispec or Report logic within a Specification or Business Segment. Global Setup Items can only be defined within Global Setup Data Blocks. *See* Setup Data Item, Group Global Setup Data Item.

GNT

The file extension for Micro Focus “generated” (native instruction set) COBOL code. Files in this format have intermediate compilation and execution times. *Contrast with* INT.

graph

Pictorial representation of a number of Elements and the relationships between the Elements. Provides access to all information supporting the represented Elements.

Group Global Setup Data Item

A Global Setup Data Item that is a concatenation of a number of other Global Setup Data Items. *See* Global Setup Data Item.

group query

For Ad Hoc Inquiry, an *SQL format query* whose output lines are for groups of related records rather than individual records. The group is defined by including a GROUP BY clause, or by including group functions such as MAX, MIN, SUM, and AVG in the *select list*.

Group Setup Data Item

A Setup Data Item that is a concatenation of a number of other Setup Data Items. *See* Setup Data Item.

H

Home position

Top left corner of a character-based screen. Often used as the position from which commands are entered.

Home screen

For Ad Hoc Inquiry, the base screen from which the main functions are chosen.

host

Computer on which an Enterprise Application Environment application is running.

Host List

A list of valid host computers that can be used by a Generate Set.

Host Name

The name of the specific host to be used for a function.

Host Specification

The name of the target host to be used for the process, its TCP/IP or network address, and host type. *See* Host Name.

HUB

The facility that controls external Automatic Entries between Enterprise Application Systems. *See* Automatic Entry, external Automatic Entry.

HUB Background Run

For OS 2200 Systems, a background run that processes external Automatic Entries. There is one Background Run for each Runtime. *See also* HUB.

HUB Listener

This service listens for and accepts HUB connection requests from other Enterprise Application user systems.

I

IC

Interim Corrections (ICs) are used to distribute new features, fixes, customer requests and newly validated versions of support software. *See also* CHG.

ICP

See Initial Control Program (for OS 2200 based Systems) and Ispec Control Program (for UNIX based Systems).

import

To copy application files into the directory where they will be deployed. When importing files from another Windows server, the import operation also decompresses the export file into its constituent application files.

Initial Control Program (ICP)

The program that controls the generated Ispec subprograms in OS 2200 based Systems. *Contrast with* COMSTP Program (for MCP based Systems) and Ispec Control Program (for UNIX based Systems).

Insertable Global Logic

A logic sequence and/or a screen layout that can be copied into any number of Ispecs and Reports. *Contrast with* Performable Global Logic.

Installation Guide

The manual that contains instructions for installing and configuring Enterprise Application Systems on a particular host type.

INT

The file extension for a file in Micro Focus interpretive code format. Files in this format compile quickly, but are slow to execute. *Contrast with* GNT.

interdatabase access

The accessing of the database of an OS 2200 based System by another OS 2200 based System.

internal Automatic Entry

An Automatic Entry from within the same Enterprise Application System. *See* Automatic Entry. *Contrast with* external Automatic Entry.

interrogation point

For Ad Hoc Inquiry, a regular interval during query execution when the status of that query is written to the database.

Ispec

(1) A collective term for Components and Events. In Enterprise Application Environment, an Ispec models an entity or activity in the real world. An Ispec also specifies the user interface, the processing rules, and the database structure to be used to represent it in the deployed user system. (2) A contraction of the term *Interface Specification*. (3) *See also* Component, Event.

Ispec Control Program (ICP)

The program that controls the generated Ispec subprograms in Enterprise Application UNIX based Systems. *Contrast with* COMSTP Program (for MCP based Systems) and Initial Control Program (for OS 2200 based Systems).

K

Keyword

A Data Item that has multiple data entry fields and Display items. *See also* Keyword Term Item, Keyword Term Display.

Keyword Term Display

The Display Item associated with a Keyword. *See* Keyword.

Keyword Term Item

The Data Item associated with a Keyword. *See* Keyword.

L

language

A natural language in which text components of the System can appear. The default language is usually English, but can be any language you chose. An application can have up to 14 languages concurrently installed. Users, or the software itself, can choose to display screens, Reports, or prompts in any one of the installed languages.

LINCTEMP

The temporary directory. Its location is defined by the environment variable *LINCTEMP* and defaults to *<LINC_DIRECTORY>\bin*.

LINQINQ

A user account with read-only privileges used to perform inquiries against Enterprise Application Control and User tables.

literal

A literal is a value used directly by a System, without requiring any named storage area. A literal may also be used in many logic commands in place of Data Items, and to define the characteristics of Setup Data Items.

Local Data Dictionary

A data dictionary that controls the use of Data Items within a single Specification. *See also* Data Dictionary, Global Data Dictionary.

lock

Inhibit other users from reading and/or updating a specific record or window. Developer has three forms of locking:

- An automatic soft lock that is set up whenever you attempt to modify an Object. This lock is released when the transaction ends or you cancel the operation.
- A manual lock that emulates the behavior of the host environment resource locking.
- If Version Control is installed, a Version lock that is set when an Object is checked out of the Version Control Bank.

locked query

For Ad Hoc Inquiry, a query that is not available for editing or running because it is currently queued for execution, or its execution has terminated abnormally, or it is being edited.

logic

Series of commands defined using the Editor that can be executed by an Object such as a Component or an Element.

logical printer

A logical printer is a printer name specified in a Report. You must map a logical printer to a physical printer before printing.

logical reorganization

Transforms the data from the existing physical database schema into the format of the newly developed logical schema in memory each time the data is accessed. File formats are not altered. Allows a new database schema to be run against the existing old database schema prior to a physical reorganization.

logically deleted

A term used to describe the status of an Ispec record that has been deleted from a database as far as all processes are concerned, but is physically still present.

Long name

Long names of Objects are up to 30 characters and allow double-byte characters, such as Kanji, to be used for an Object's name. They also allow you to give Objects more meaningful names. Long names must be used in conjunction with shorter names which can be used by the host environment.

LSM

See Security Module.

M

Main Logic

User-specified Ispec logic that is executed after any Pre-LINC logic and automatic editing and validating sequences have been executed.

MAINT

(1) A System Data Item, stored with each record of every Standard, Table, and Automaint Memo Component, that indicates whether that record is added, changed or deleted. (2) In mainframe Systems, a screen field for every Standard, Table, and Automaint Memo Component used to indicate the database maintenance action request.

MCP environment

The component of the A Series or ClearPath HMP NX machine that runs the MCP operating system.

Memo Component

A Component type that has no Ordinate, and which is used to store data of a memorandum nature. *See* Component, Automaint Memo Component.

Memo Data

Memo-type, optional Ispec data that is stored separately to the primary Ispec records to assist with the optimization of disk usage. Valid for MCP based Systems only. *See* Ispec.

menu format query

A query created using the various selection screens of Ad Hoc Inquiry. *Contrast with* SQL format query.

Message Translation Utility (MSGTRANS)

A part of the Multilingual System (MLS) that co-ordinates the output of translated messages.

MLA

See Multiple Language Mode.

Model

Database containing details of Business Segments held by Developer. A synonym for Repository.

Model Directory

A graphical representation of the hierarchical structure of the Business Model. An essential part of the Developer navigation facilities.

Multilingual System (MLS)

An MCP based facility that provides data structures and associated access methods to store and retrieve the translatable text used by software. *See also* Message Translation Utility.

multiple language facilities

A collective term for the facilities that enable an Enterprise Application System to be translated and used in up to fifteen different languages, from a single Specification.

Multiple Language Mode (MLA)

A facility in mainframe Enterprise Application Environment that enables a language name or number to be defined. *See* multiple language facilities.

Multiple Ordinate Memo Component

See Automaint Component.

N

navigation tree

In the Administration GUI, a depiction of the Windows domains, servers, environments, databases, and Systems. The navigation tree is updated as the user makes changes to the environment.

NOF

A facility for interfacing to and from an Enterprise Application System using Non-Formatted Input/Output, a message-oriented interface between an Enterprise Application user system, and usually, an external system and terminal. The interface to Graphical Interface Workbench uses NOF.

NOFORM

See NOF.

Non-Formatted Input/Output (NOF)

See NOF.

O

OBJ

The file extension for a file in standard object code format. Files in this format compile slowly, but execute quickly. Files in .obj format cannot be executed unless bound.

Object

Collective term for a Component, Event, Inquiry, Report, Global Logic, or Profile.

object files

In Runtime for Windows Operating Systems, the .obj files that result from compiling source files. Object files usually require linking to other program modules before they can be executed.

Object Packaging

On UNIX Systems, Object Packaging allows the creation of executables from supplied Enterprise Application object files, the installation of Enterprise Application software, the installation of ICs, and the maintenance of your Enterprise Application environment. It also allows the creation of new object and executable packages which can include supplied *ICs*. Packages can be created for distribution and as backups.

Objects

(1) Objects are one of two methods of distributing Enterprise Application software to customers. They are the set of object files which must be built into executables on your site before Enterprise Application Environment can be run. Object files are built using *Object Packaging*. Contrast this with *Executables*. (2) In Runtime for the Windows 2000 Operating System Administration Client, anything that appears in the navigation tree, including servers, client machines, environments, databases, and user systems.

Offline interface

A facility for passing batches of high-volume transactions from external systems into an Enterprise Application user system.

OLTP

See Online Transaction Processing.

OLTP Buffer Definition file

An OLTP Buffer Definition file holds the data format for communicating with external OLTP Servers.

OLTP View Ispec

In the host environment, an OLTP View Ispec (OVI) is used to store an OLTP View Description file. This file holds the data format for communicating with external OLTP Servers.

On Change Statistical Routine

A statistical routine (for example, AVERAGE;) that can be performed on a Data Item as part of an ON.CHANGE; logic command.

Online Transaction Processing (OLTP)

A generic method for transferring transactions between Systems.

Oracle

Relational database management system used as the underlying database software for Enterprise Application Systems based on the UNIX or Windows operating systems. Oracle is produced by Oracle Corporation.

Oracle Hints

A facility that allows users to add information to SQL SELECT statements. Correct use may improve database performance, but incorrect use may reduce database performance. Using Oracle Hints in an SQL statement overrides the Oracle Optimizer. Users are able to specify:

- Which index is used in the database access operation
- Whether the index should be read in reverse order

Oracle SID

An occurrence or instance of an Oracle database. The Oracle SID is a 1 or 2 character identifier. An Oracle term for a logical storage area that houses user systems and control tables and may contain one or more databases. The SID identifies a particular database occurrence. *See also* tablespace.

Oracle tablespace

A discrete part of an Oracle SID. Tablespaces in the same Oracle SID can be stored on different physical devices.

Ordinate

The Data Item of a Component that acts as the unique identifier for a record. The access path to individual Component records.

P

pack

A definition of the location where part of a user system or an environment database is stored. This may be either a tablespace or a directory path.

Pack Association

The Pack Association specifies the location in which the files of the generated System will be stored on the target host computer.

Pack View

The Pack View in Developer lists all the packs used in the Systems options of the currently open Model. This specifies the Pack associations, or the location in which that part of the System or database is stored. *See* Pack.

Painter

A Developer facility used to define an Ispec screen layout or Report Frame layout.

patch

A set of changes to a Specification made over a period of time by users signing on to Enterprise Application Environment under a specific Patch identifier value.

Patch identifier

A number used to identify and group changes to a Specification by one or more users. Applies when the Patch identification option is selected for that Specification. *Contrast with* Change identifier.

Performable Global Logic

Logic that can be executed by any number of Ispecs and Reports. *Contrast with* Insertable Global Logic.

physical delete

Removal of a record from a database or repository.

physical reorganization

Copies and transforms the physical data from the existing database into a form which matches the new database schema.

population

Also called Expected Number. The maximum number of records expected to be stored in the database for an Ispec.

Pre-LINC Logic

User-specified Ispec logic that is executed before the Ispec Main logic.

Pre-Screen Logic

A section of user-specified Ispec logic that is executed before the screen layout is displayed on the terminal.

Product Menu Utility

Product Menu is a menu-driven utility to access Object Packaging features. See *Object Packaging* for details of the functions that can be performed with Product Menu.

Profile

(1) An index to a specified selection of records. (2) A method of providing access to just those records that are required to perform a specific function. (3) A functional view of the database.

Profile Data

A Data Item that is physically stored in a Profile as well as in its associated Ispec. *See also* Profile.

Profile Ordinate

The Data Item by which a Profile accesses individual records. A Profile can have several Profile Ordinates. *See also* Profile.

Q

query class

See access class.

Query Compiler

The Ad Hoc Inquiry facility that validates an entered query and transforms it into a standard internal code and format. *See also* query transformation.

Query Front End

The Ad Hoc Inquiry user interface that allows a query to be entered.

Query Optimizer

The Ad Hoc Inquiry facility that selects the most efficient way to access the database for a query.

Query Output Handler

The Ad Hoc Inquiry facility that enables users to manipulate query output.

Query Processor

The Ad Hoc Inquiry facility that interprets the query code and runs the query against the database.

Query Processor report

For Ad Hoc Inquiry, the program that performs the functions of the Query Processor.

query transformation

For Ad Hoc Inquiry, the process performed by the Query Compiler to convert a validated query into a standard internal form.

queued query

For Ad Hoc Inquiry, a query waiting for resources to begin execution.

R

recovery

The process of restoring database files from a backup and performing roll-forward recovery from a current transaction log file. Failed Reports must be restarted manually, as they are not automatically restarted by the recovery process.

relation name

For Ad Hoc Inquiry, an SQL term equivalent to a Component, the Event file, or an Event Profile. Also called a table.

relationship

An association between two Elements. Relationships may be a result of design (for example, added using a graph) or logic (for example, adding a FLAG; command to an Ispec).

relative day number

A date expressed as the number of days since January 1 of the base year. The relative day number of January 1 of the base year is zero. *See also* base year.

Remote Access to Enterprise Application

The Remote Access server resides on the Runtime host and provides the basis for communication between user Views and Enterprise Application Systems. Can be used with Graphical Interface Workbench as well as Component Enabler Viewer applications. It performs services previously provided by WDP. Formerly known as RATL Server. *See* Component Enabler, Workstation Driver Program (WDP).

reorganization

The process by which the physical database for a System is updated to match the logical definition held in the Repository. *See also* logical reorganization and physical reorganization.

Report

Part of a Specification generated and used to produce output or to carry out specialized batch processing of a database. Consists of Report Frames and Report Main logic, and a number of options that define the operation and output of the Report.

Report Frame

Layout that contains display text and Data Items, and some associated logic that is part of a Report. The logic is executed (often to define the Data Items in the layout) and then the layout is included in the Report output. Report Frames are accessed as single units from the logic of Report Frame or from Main logic of the Report.

Report Group

A set of Reports that are generated as a group. All the Reports assigned to the group are generated by selecting the Report Group for generation.

Report Initiation Program (RIP)

An MCP based utility that enables you to execute a line printer Report or terminal printer Report in batch mode.

Report Output Control System (ROC)

A utility that provides control over the output of a Report.

Repository

Database containing details of Business Segments (Specifications) held by Developer. A synonym for Model.

reserved words

Words reserved for system use. Each architecture has its own set of reserved words.

resource locking

A facility of the host environment that prevents access to parts of a Specification by more than one user at a time.

ROC Background Run

For OS 2200 Systems, a background run that identifies Report output requests and spools the output to the required print queue. There is one ROC Background Run for each Runtime.

rollback segment

In Oracle, a segment that contains an image of data before committing to a transaction. Used for rolling back uncommitted transactions, read consistency, and recovery, as a manual alternative to automatic undo management.

row

For Ad Hoc Inquiry, an SQL term equivalent to a Component record or an Event record.

RPC

Remote Procedure Call, a Windows protocol that runs background services that are necessary for the operation of the Administration Client and Runtime for Windows Operating Systems.

Runtime

A collective term for the software programs required to operate, control and audit an Enterprise Application System and its Database.

Runtime platforms

The platforms on which applications developed and built using Enterprise Application software runs: Windows 2000, OS 2200, MCP, and the UNIX operating system.

S

schema

Database structure.

screen

In Developer, a screen is a means of entering a transaction. Screens are usually designed for a single purpose and so each System function will usually have a specific screen. Screens may have associated logic.

search expression

For Ad Hoc Inquiry, a logical expression used in a search list to define records to be included in query output.

search list

For Ad Hoc Inquiry, the part of a query that determines whether records will be included in the output.

security access

For Ad Hoc Inquiry, full access to all queries.

security class

For Ad Hoc Inquiry, a type of user access to queries. *See* access class.

security level

A value from 0 through 9 that is assigned to a specific user or terminal to restrict access to functions.

Security Module (LSM)

A utility which provides security facilities for Enterprise Application environments.

select list

For Ad Hoc Inquiry, the part of a query that defines the Data Items to be included in the output.

Setup Data Array or Global Setup Data Array

Use a Setup Data array or Global Setup Data array to store multiple values. An individual value is accessed by specifying values for the indexes of the array. Indexes are numeric Setup Data Items or Global Setup Data Items.

Setup Data Item

A Data Item used in memory only for data manipulation by an Ispec or Report. *See also* Global Setup Data Item, Group Setup Data Item.

Shadow Report

Part of a Report used to produce additional output separate from that produced by the primary functions of the Report.

SID

The two-character identifier of an Oracle database. In Runtime for Windows Operating Systems, *database*, *Database ID* and *Oracle SID* are synonymous.

Sleeping Report

A Report that, by the use of the SLEEP; logic command, stops executing for a predetermined number of seconds or until reactivated by your runtime System.

SOAP

Simple Object Access Protocol. A standard for encoding XML messages for transmission over HTTP. It is a lightweight protocol for the exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses

SOLAR

For OS 2200 Systems, a product used to install Enterprise Application Software.

sort expression

For Ad Hoc Inquiry, a logical expression used in a sort list to define output sort order.

sort list

For Ad Hoc Inquiry, the part of a query that defines the order of the output.

Specification

(1) The result of specifying a Business Model using Developer. (2) The source from which an Enterprise Application System is generated.

spooler

See Terminal Printer Spooler.

SQL format query

An SQL query entered in free format in Ad Hoc Inquiry. *Contrast with* menu format query.

SQL Server

A relational database management system that is optionally used as the database software for Enterprise Application Systems based on Windows operating systems. SQL Server 7.0 is a product of Microsoft Corporation. *See also* Oracle.

Standard Component

A Component type that requires a single Ordinate. *See also* Component.

Standard Report

A specific type of Report that uses the Report Output Control (ROC) System, and whose output is written to the database. Subsequent use of the output is determined by the user. *See* Report Output Control System. *Contrast with* Direct Report.

station

(1) In Enterprise Application Environment terminology, a general term for a method used to identify users. (2) For UNIX based Systems, station equates to user name, while for OS 2200 based Systems, station equates to userid.

status line

A line on a screen for the display of system messages.

structure

(1) In the host environment, part of a Specification that can be selected (for example, for printing, copying, or extraction). (2) For Ad Hoc Inquiry, part of an Enterprise Application Database that can be the subject of a query.

structure list

For Ad Hoc Inquiry, the part of a query that defines the structures to be included.

Sub Items

A Global Setup Data Item that has been defined for a Group Global Setup Data Item or a Setup Data Item that has been defined for a Group Setup Data Item.

Sub Scripts

A Data Item that has been defined as an index for either a Global Setup Data Array or a Setup Data Array.

Supervisor

A facility of the host environment enabling administration-level functions.

System

In Builder each System consists of a configuration of elements and the options assigned to these elements. The options define the generation options for the host-based environment on which the System will be deployed.

System command

A command to assist in the operation of a System or Report. Formerly known as a colon command.

System Data Item

Data Item that is automatically part of your Specification. Used for accessing or setting parameter or control-type data in logic. Some System Data Items are stored automatically in your Enterprise Application Database (for example, MAINT).

System Directory

A graphical representation of the hierarchical structure each System configured for a Business Model. It lists the current and previous configurations for each System, showing the Elements assigned to the System and the processing options defined for them. The System Directory provides the basis for the build process.

System Element

Collective term for the individual parts of the System. An Element may be any of the following: Business Segment, Bundle, Ispec, Global Logic, Profile, Report Group, or Report.

System Specification

A specification that represents the configuration of Elements within a deployed host System, and the generation options required to generate a set of executables for the specified host environment. *See* Generate Set.

T

table

For Ad Hoc Inquiry, an SQL term equivalent to a Component, the Event file, or an Event Profile. Also called relation name.

Table Component

A Component type that is used to store low-volume, frequently accessed data, such as code tables. *See also* Component.

tablespace

Discrete part of an Oracle SID. Tablespaces in the same Oracle SID can be stored on different physical devices.

target host

The name of the specific host on which a generated Enterprise Application System will be deployed. *See also* Host Specification.

Teach screen

A user-written screen display of information about an Ispec, intended to provide online help for end users at runtime.

Terminal Printer Spooler (TPS)

An MCP based Enterprise Application Utility that enables users to manipulate the output of a Report to terminal printers.

TPS

See Terminal Printer Spooler.

Trace

A facility to monitor the execution of logic of an Ispec or Report in your Enterprise Application System.

transaction log file

File to which before and after images of all Developer transactions are written for recovery and restart purposes.

Translation Screen

A copy of an Ispec screen that is modified to create a secondary language version.

U

undo tablespace

In Oracle, a tablespace used for storing information that can be used to undo, or roll back, changes to the database when necessary.

Usage Input

An Ispec attribute that enables a Data Item to be entered or displayed on the screen, accessed by logic, but not stored in the database.

Usage Input Ispec

An Event or Memo Component that is not output to the database, and where Data Items default to Usage Input. (Individual Data Items can be set to Usage Inquiry.)

Usage Input-Output

An Ispec attribute that enables a Data Item to be entered or displayed on the screen, accessed by logic, and stored in the database.

Usage Input-Output Ispec

Ispec that is both Usage Input and Usage Output.

Usage Inquiry

An Ispec attribute that enables a Data Item displayed on the screen, but not to be entered on the screen, nor written to the database.

Usage Output

An Ispec attribute that enables a Data Item to be written to the database and accessed by logic, but not entered or displayed on the screen.

Usage Output Ispec

An Ispec that is written to the database, but does not have a screen display.

user system

An instance of a Specification. Generally refers to a generated and deployed Enterprise Application user system.

user system tables

Tables that contain end user data (Ispec data) of a specific Enterprise Application user system.

V

Validation IC/CHG

Validation ICs and CHGs are a special type of ICs and CHGs which provide support for newly validated versions of Enterprise Application Environment support software such as Oracle, COBOL, and Tuxedo. *See* IC *and* CHG.

value expression

For Ad Hoc Inquiry, a logical expression used in an SQL format query to define a value, and made up of Data Items, literals, operators, and functions.

value logic

Logic associated with a dictionary item that determines acceptable values for that item.

version control

A method for identifying different versions of an Enterprise Application System by version number, description, and generation date.

Version Control

An optional feature of Developer that provides source control of versionable Enterprise Application objects such as Model definitions, data dictionary items, Ispecs, Profiles, and Reports.

W

watch

A feature of the Developer Test Debugger that monitors, or 'watches', the state of a Data Item. When logic execution is stopped during debugging, the value of the watch item is displayed in the Watch Window.

Web Service

A Web Service is an application module, which can be invoked remotely from another application using the infrastructure of the Internet/Intranet. In the context of Enterprise Application Environment, a Web Service is a wrapper module, which receives messages based on the Simple Object Access Protocol (SOAP) protocol. It calls the appropriate Ispec using Component Enabler and returns the Ispec fields as an XML message.

Wildcard

Represents relationships with objects outside the Business Segment, or outside the Model. For example, a flat file that is read or extracted, or an external system called by the Enterprise Application System.

Workstation

In a client-server environment, a PC or workstation that is connected to a server, usually by a network. The workstation runs the client portion of client-server software.

Workstation Driver Program (WDP)

A server-resident component of Runtime for Windows Operating Systems. Its function is to pass data between the Graphical Interface Workbench workstation and the application on the runtime host. It also converts screen definition files during the direct generation of an application.

Workstation Driver Program (WDP) listener

This service listens for and accepts connection requests from Graphical Interface Workbench clients.

Workstation listener port

A Windows TCP/IP port assigned to the Workstation Driver Program (WDP) listener or a particular environment. This value must be changed to a unique number for each user system environment during installation.

WSDL

Web Service Description File. A standard for an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.

X

XML

Extensible Markup Language (XML) is a flexible, universal file format for creating data for common information formats and sharing both the format and the data on the Web, intranets, and elsewhere. An XML file contains data in identified fields that are used by XSL stylesheets to present the data in the XML file.

XSL

Extensible Stylesheet Language (XSL) is used for creating stylesheets that describe how data sent over the Web using XML is to be presented to a user. XSL gives developers the tools to describe exactly which data fields in an XML file to display, and exactly how and where to display them.

Index

UNISYS

78616562-003