Interaction Host Recorder

Printable Help

Interactive Intelligence Customer Interaction Center® (CIC)

2016 R1

Last updated December 15, 2015
(See Change Log for summary of changes.)

Abstract

This document describes Interaction Host Recorder, the CIC application for creating host profiles. Host profiles are the scripts that Host Recorder uses to navigate terminal emulation screens.

Interactive Intelligence, Inc.
7601 Interactive Way
Indianapolis, Indiana 46278
Telephone/Fax (317) 872-3000
www.ININ.com

Copyright and Trademark Information

Interactive Intelligence, Interactive Intelligence Customer Interaction Center, Interaction Administrator, Interaction Attendant, Interaction Client, Interaction Designer, Interaction Tracker, Interaction Recorder, *Interaction Mobile Office, Interaction Center Platform, Interaction Monitor, Interaction Optimizer,* and the "Spirograph" logo design are registered trademarks of Interactive Intelligence, Inc. *Customer Interaction Center, EIC, Interaction Fax Viewer, Interaction Server, ION*, *Interaction Voicemail Player, Interactive Update, Interaction Supervisor, Interaction Migrator,* and *Interaction Screen Recorder* are trademarks of Interactive Intelligence, Inc. The foregoing products are ©1997-2015 Interactive Intelligence, Inc. All rights reserved.

Interaction Dialer and *Interaction Scripter* are registered trademarks of Interactive Intelligence, Inc. The foregoing products are ©2000-2015 Interactive Intelligence, Inc. All rights reserved.

Messaging Interaction Center and *MIC* are trademarks of Interactive Intelligence, Inc. The foregoing products are ©2001-2015 Interactive Intelligence, Inc. All rights reserved.

Interaction Director is a registered trademark of Interactive Intelligence, Inc. *e-FAQ Knowledge Manager* and *Interaction Marquee* are trademarks of Interactive Intelligence, Inc. The foregoing products are ©2002-2015 Interactive Intelligence, Inc. All rights reserved.

Interaction Conference is a trademark of Interactive Intelligence, Inc. The foregoing products are ©2004-2015 Interactive Intelligence, Inc. All rights reserved.

Interaction SIP Proxy and *Interaction EasyScripter* are trademarks of Interactive Intelligence, Inc. The foregoing products are ©2005-2015 Interactive Intelligence, Inc. All rights reserved.

Interaction Gateway is a registered trademark of Interactive Intelligence, Inc. *Interaction Media Server* is a trademark of Interactive Intelligence, Inc. The foregoing products are ©2006-2015 Interactive Intelligence, Inc. All rights reserved.

Interaction Desktop is a trademark of Interactive Intelligence, Inc. The foregoing products are ©2007-2015 Interactive Intelligence, Inc. All rights reserved.

Interaction Process Automation, Deliberately Innovative, Interaction Feedback, and *Interaction SIP Station* are registered trademarks of Interactive Intelligence, Inc. The foregoing products are ©2009-2015 Interactive Intelligence, Inc. All rights reserved.

Interaction Analyzer is a registered trademark of Interactive Intelligence, Inc. *Interaction Web Portal,* and *IPA* are trademarks of Interactive Intelligence, Inc. The foregoing products are ©2010-2015 Interactive Intelligence, Inc. All rights reserved.

Spotability is a trademark of Interactive Intelligence, Inc. ©2011-2015. All rights reserved.

Interaction Edge, CaaS Quick Spin, *Interactive Intelligence Marketplace, Interaction SIP Bridge,* and *Interaction Mobilizer* are registered trademarks of Interactive Intelligence, Inc. *Interactive Intelligence Communications as a Service℠*, and *Interactive Intelligence CaaS℠* are trademarks or service marks of Interactive Intelligence, Inc. The foregoing products are ©2012-2015 Interactive Intelligence, Inc. All rights reserved.

Interaction Speech Recognition and *Interaction Quality Manager* are registered trademarks of Interactive Intelligence, Inc. *Bay Bridge Decisions* and *Interaction Script Builder* are trademarks of Interactive Intelligence, Inc. The foregoing products are ©2013-2015 Interactive Intelligence, Inc. All rights reserved.

Interaction Collector is a registered trademark of Interactive Intelligence, Inc. *Interaction Decisions* is a trademark of Interactive Intelligence, Inc. The foregoing products are ©2013-2015 Interactive Intelligence, Inc. All rights reserved.

Interactive Intelligence Bridge Server and *Interaction Connect* are trademarks of Interactive Intelligence, Inc. The foregoing products are ©2014-2015 Interactive Intelligence, Inc. All rights reserved.

The veryPDF product is ©2000-2015 veryPDF, Inc. All rights reserved.

This product includes software licensed under the Common Development and Distribution License (6/24/2009). We hereby agree to indemnify the Initial Developer and every Contributor of the software licensed under the Common Development and Distribution License (6/24/2009) for any liability incurred by the Initial Developer or such Contributor as a result of any such terms we offer. The source code for the included software may be found at http://wpflocalization.codeplex.com.

A database is incorporated in this software which is derived from a database licensed from Hexasoft Development Sdn. Bhd. ("HDSB"). All software and technologies used by HDSB are the properties of HDSB or its software suppliers and are protected by Malaysian and international copyright laws. No warranty is provided that the Databases are free of defects, or fit for a particular purpose. HDSB shall not be liable for any damages suffered by the Licensee or any third party resulting from use of the Databases.

Other brand and/or product names referenced in this document are the trademarks or registered trademarks of their respective companies.

DISCLAIMER

INTERACTIVE INTELLIGENCE (INTERACTIVE) HAS NO RESPONSIBILITY UNDER WARRANTY, INDEMNIFICATION OR OTHERWISE, FOR MODIFICATION OR CUSTOMIZATION OF ANY INTERACTIVE SOFTWARE BY INTERACTIVE, CUSTOMER OR ANY THIRD PARTY EVEN IF SUCH CUSTOMIZATION AND/OR MODIFICATION IS DONE USING INTERACTIVE TOOLS, TRAINING OR METHODS DOCUMENTED BY INTERACTIVE.

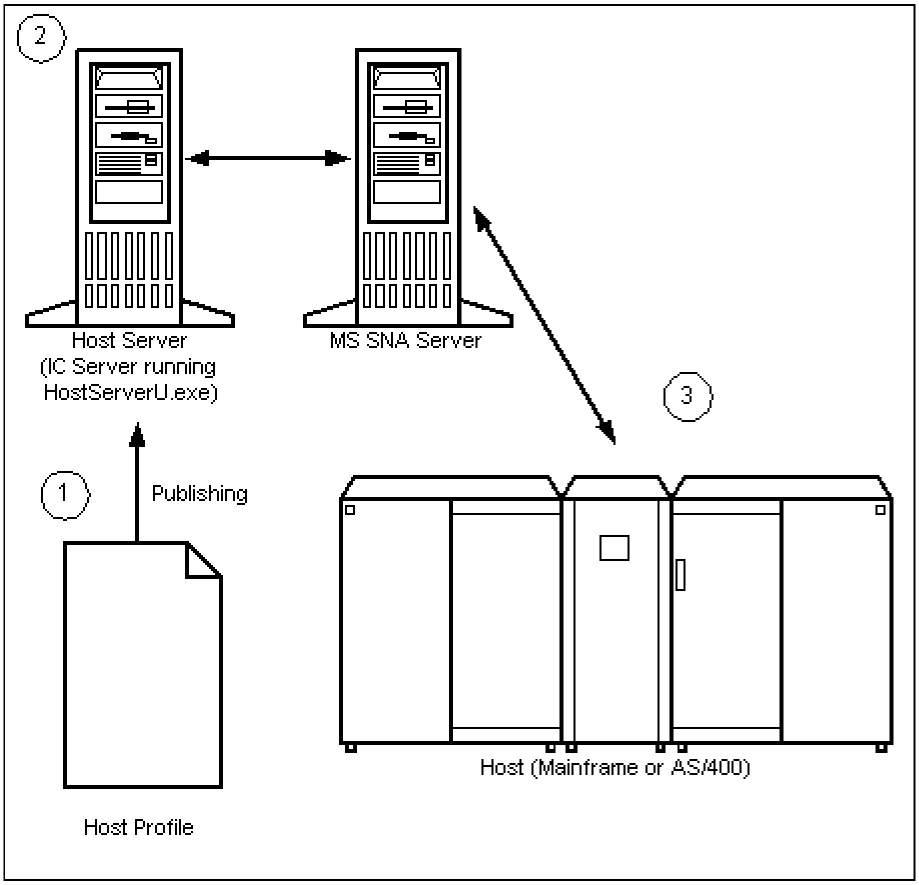
Interactive Intelligence, Inc.
7601 Interactive Way
Indianapolis, Indiana 46278
Telephone/Fax (317) 872-3000
www.ININ.com

Table of Contents

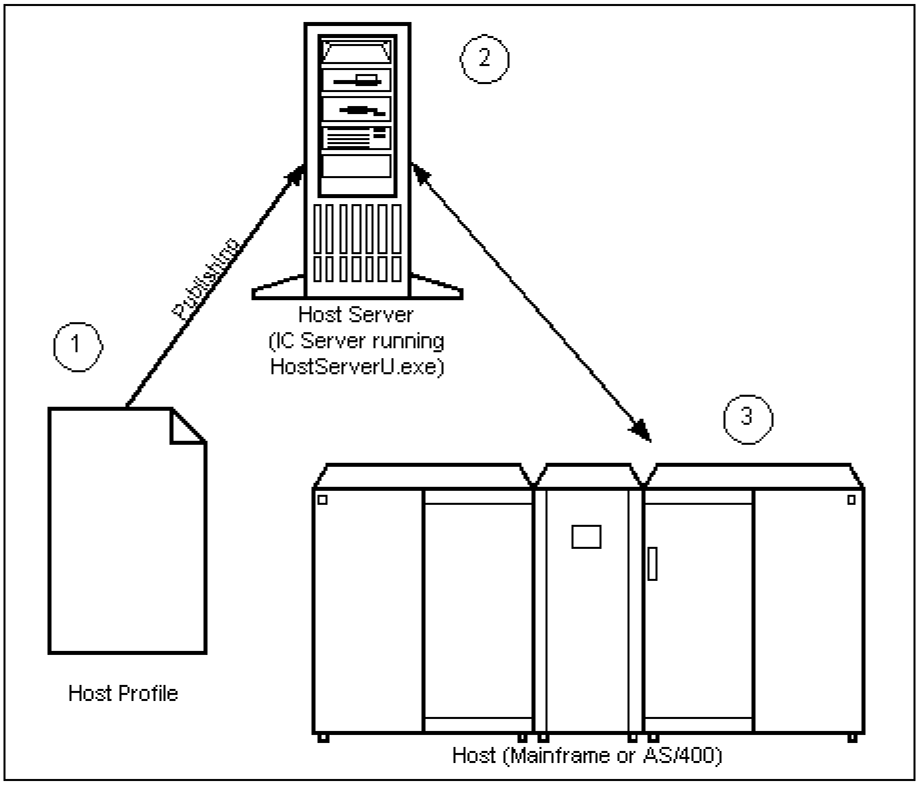
Overview

Welcome to Interaction Host Recorder, the IC application for creating host profiles. Host profiles are the scripts Host Server uses to navigate terminal emulation screens. The host profiles you publish to the Host Server can be used by the Host tools in handlers, as shown in the following two figures:

Configuration with MS SNA Server



Configuration with Direct Telnet Connection



1. A host profile that you created in Interaction Host Recorder is published to the Host Server. The profile contains information about connecting to a host, moving between screens (transitions), and areas in the screens where information can be read and written.
2. Host tools in handlers send commands to the Host Server. These tools can request that Host Server use the profile to open a connection to the host, navigate to certain screens, and read and/or write data. HostServerU.exe executes the commands sent by the host tools. The connection is opened and a terminal emulation is maintained on the Host Server. The Host Server can maintain thousands of these connections simultaneously.
3. The mainframe host does what the Host Server tells it to do. As far as it knows, the Host Server is just another human user typing commands.

As far as the logic involved in the host connection is concerned, connections to the SNA Server can be considered to be the same as connections to the host in a Telnet configuration.

See Introduction to profiles for an overview of the how you can create hot profiles in Interaction Host Recorder.

Introduction to profiles

Interaction Host Recorder is primarily a tool for building and editing host profiles, the scripts that allow handlers to send and receive data from a host session. This topic offers an overview of one profile building method, but once you are familiar with the process you can perform these steps in a different order.

Before you begin

Before you can begin creating host profiles make sure you have login and password information that the handlers will need to connect with the host.

1. Configure the connection properties

   Before you can start a host session you must configure connection properties such as host address, type, and several more. The online help for the Connection dialog explains the options. Once properly configured, you can connect to the host and open the first screen.

2. Define the screen

   After connecting, you should define the current screen. You must name the current screen before you can create validation rules, user defined fields, or transitions. Handlers cannot act upon screens that are not defined.

3. Create validation rules for the new screen

   Validation rules are the distinguishing characteristics of a specific named screen. You can define a screen by its screen ID, the appearance of certain text in a certain location, or several other characteristics. The online help for the Validation Rules list explains the different kinds of validation rules.

4. Creating user defined fields

   User defined fields are the areas on the screen where handlers can read and write information. There are eight types of user defined fields that allow you to define a set of coordinates, or a position relative to some other element on the screen. The online help for User Defined Fields explains your options. You can create user defined fields on any named screen.

5. Creating Transitions

   After defining one or more screens and setting the validation rules for those screens, you will probably want to create transitions between the screens. Transitions are the collection of instructions (transition commands) used to move from one screen to another. Interaction Host Recorder has a recording feature that allows you to automatically record transitions, or you can create and edit them manually.

6. Publishing Host Profiles

   When your host profile contains the defined screens, user defined fields, and transitions that you need, you can publish the host profile to the Host Server. The Host tools in handlers can then tell the Host server to run the profiles and perform one or more transitions within the host profile.

Command Line Parameters

The following command line parameters can be used when starting Interaction Host Recorder:

`/s={servername}`

`/notifier={servername}`

`/server={servername}`

IC server for publishing scripts to (if other than default)

`/userid={userid}`

User ID for publishing (if other than default)

`/password={password}`

Password for above User ID

`/nopublish`

Disables script publishing operations (prevents application from locking up for about 2 minutes when publishing attempted without access to an IC server).

`/trace={type}`

Sets level of tracing added to the trace buffer and shown in the trace history dialog.

{type}:

none No tracing

error Trace only errors

warning Trace errors and warnings

status Default, trace errors, warnings and status information

note All of the above including additional verbose information about execution process.

Note: The parameter keywords are not case sensitive.

Connections

Connecting to a host

You should connect to a host before creating or editing a host profile. Once you have connected to the host, the screen appears as a terminal emulation in which you can navigate and create the host profile.

If you are creating a new profile:

After starting Interaction Host Recorder, open a new profile by clicking the New button on the toolbar. The Connection Properties dialog appears. It is in this dialog that you must configure the connection settings, including the host type, interface type, and host address. The Connection Properties dialog help contains detailed information on configuring the connection. Your connection settings will be saved when you save the profile.

After configuring the connection properties, click the Connect button to establish a connection with the host.

If you are editing an existing profile:

If you are editing an existing profile, connection properties already exist for that profile. You can change these connection settings by choosing **Connection Properties...** from the **Host Menu**. The Connection Properties dialog appears. You should reconnect to the host after changing the connection settings for those changes to take effect.

Once the connection settings are correct, click the Connect button to establish a connection with the host.

After you have connected to the host, you can set up validation rules, user defined fields, and transitions.

Connection Properties

Connection properties are the settings to open a connection to the host. In this dialog you'll configure the host type, address, display options, and several other settings. You must configure the connection properties for each new Host Profile that you create.

Once you've configured the parameters in this notebook, you can open the terminal emulation by clicking the Connect button.

How to access the Connection Properties dialog

General Page

Name

The name of the host.

Host Type

Sets the type of host you want to connect to. This release supports Mainframe and AS/400 hosts. The choice you make here determines what configuration options are available on the Attributes page.

Interface

Shows or sets the host communications interface used to connect to the host. Use this drop-down list to determine which interface is currently selected or select an interface to connect with the host. For both Mainframe and AS/400 host types, you can select TN 5250 (Telnet) or MS SNA Server.

Display Mode

Shows or sets the width and height for the display of the terminal emulation. If you choose the wrong size, the terminal emulation may appear disorganized. If you have problems with the terminal emulation display, try changing this value.

Timeout

Shows or selects the number of seconds to wait for a connection before stopping the connection process. The connection is not retried after this time is exceeded.

Driver

Both Recorder and Server driver are unique for the Host Type/Interface pair. For example, the Recorder driver for Mainframe TN 3270 is I3WdExpMFDriver. If you accidentally delete these settings, see Default Connection Drivers for the default settings.

Caution: These drivers are predefined during Interaction Host Recorder installation, and you should not change the default values unless instructed to do so by an Interactive Intelligence support agent.

As Default button

Saves the current settings as the default for the current Host Type/Interface pair. If you click this button, a warning dialog appears for you to confirm the action.

Attributes Page

The fields available on the Attributes page will change depending upon which Host Type/Interface pair you chose on the General page.

Click on one of the pairs below for more information on the configurable attributes for that pair:

- Mainframe TN3270
- Mainframe MS SNA Server
- AS/400 TN5250
- AS/400 MS NSA Server

Host types

Connection Attributes for AS/400 TN5250

The following fields are available on the Attributes page if AS/400 TN5250 is the Host Type/Interface pair chosen on the Connection Properties General Page.

How to access the Connection Properties dialog

Address

Numeric or textual IP host address. For example, you could type locis.loc.gov or 140.147.254.3.

Port

The port over which the connection will be made with the host. Use 23 with TN 5250.

Device

Displays the logical name used by the host to identify this session. In most cases you can leave this field blank. Use Device to assign a device name that will be used instead of the default device name. In most cases, the device name defaults to the PC System name followed by S1 (workstation) or P1 (printer). For example, when the PC System name is MSMITH, the default device name is MSMITHS1.

SSL Enabled option *

Indicates whether SSL encryption is enabled.

Authentication *

Indicates whether SSL authentication is enabled.

Licensing Server *

Sets the licensing server name for the Secure Socket Layer (SSL).

Failover Address

Provides the network connection redundancy for a host connection. This optional address allows ININ Host Server Tools to use an alternate, backup address to the host server (mainframe) in the event that the primary host address is unreachable.

Failover Port

The port to use for a specified failover address.

* IC Host Recorder does **not** natively include SSL encryption and authentication support, but if you use a recent version of NetManage OnWeb Web-to-Host (version 5.2 or greater) on your system, IC Host Recorder can take advantage of these SSL options.

Connection Attributes for AS/400 MS SNA Server

The following fields are available on the Attributes page if AS/400 MS SNA Server is the Host Type/Interface pair chosen on the Connection Properties General Page.

How to access the Connection Properties dialog

Device

Displays the logical name used by the host to identify this session. In most cases you can leave this field blank. Use Device to assign a device name that will be used instead of the default device name. In most cases, the device name defaults to the PC System name followed by S1 (workstation) or P1 (printer). For example, when the PC System name is MSMITH, the default device name is MSMITHS1.

Local LU

Your local machine name.

Partner LU

Typically a system assigned value.

Auto Sign-on option

Selecting this option causes an automatic login using User ID and User Password when the connection is made.

User ID

The user ID to use when logging in.

User Password

The password associated with the user ID above.

Caution: This password is not protected, allowing a user to view the password in the host profile (.hcs file). Use the security features in your operating system to protect the host profile from unauthorized users.

Connection Attributes for Mainframe TN3270

The following fields are available on the Attributes page if Mainframe TN3270 is the Host Type/Interface pair chosen on the Connection Properties General Page.

How to access the Connection Properties dialog

Address

Numeric or textual IP host address. For example, you could type "locis.loc.gov" or "140.147.254.3".

Port

The port over which the connection will be made with the host. 23 is the default for telnet.

Device

Displays the logical name used by the host to identify this session. In most cases you can leave this field blank. Use **DeviceName** to assign a device name that will be used instead of the default device name. In most cases, the device name defaults to the PC System name followed by S1 (workstation) or P1 (printer). For example, when the PC System name is MSMITH, the default device name is MSMITHS1.

Terminal

In most cases you can leave this field blank.

Extended Attributes

Returns or sets a value that specifies whether the TN3270 server supports 3270 extended attributes (TN3270E).

SSL Enabled option *

Indicates whether SSL encryption is enabled.

Authentication *

Indicates whether SSL authentication is enabled.

Licensing Server *

Sets the licensing server name for the Secure Socket Layer (SSL).

Failover Address

Provides the network connection redundancy for a host connection. This optional address allows ININ Host Server Tools to use an alternate, backup address to the host server (mainframe) in the event that the primary host address is unreachable.

Failover Port

The port to use for a specified failover address.

* IC Host Recorder does **not** natively include SSL encryption and authentication support, but if you use a recent version of NetManage OnWeb Web-to-Host (version 5.2 or greater) on your system, IC Host Recorder can take advantage of these SSL options.

Connection Attributes for Mainframe TN3270

The following fields are available on the Attributes page if Mainframe TN3270 is the Host Type/Interface pair chosen on the Connection Properties General Page.

How to access the Connection Properties dialog

Address

Numeric or textual IP host address. For example, you could type "locis.loc.gov" or "140.147.254.3".

Port

The port over which the connection will be made with the host. 23 is the default for telnet.

Device

Displays the logical name used by the host to identify this session. In most cases you can leave this field blank. Use **DeviceName** to assign a device name that will be used instead of the default device name. In most cases, the device name defaults to the PC System name followed by S1 (workstation) or P1 (printer). For example, when the PC System name is MSMITH, the default device name is MSMITHS1.

Terminal

In most cases you can leave this field blank.

Extended Attributes

Returns or sets a value that specifies whether the TN3270 server supports 3270 extended attributes (TN3270E).

SSL Enabled option *

Indicates whether SSL encryption is enabled.

Authentication *

Indicates whether SSL authentication is enabled.

Licensing Server *

Sets the licensing server name for the Secure Socket Layer (SSL).

Failover Address

Provides the network connection redundancy for a host connection. This optional address allows ININ Host Server Tools to use an alternate, backup address to the host server (mainframe) in the event that the primary host address is unreachable.

Failover Port

The port to use for a specified failover address.

* IC Host Recorder does **not** natively include SSL encryption and authentication support, but if you use a recent version of NetManage OnWeb Web-to-Host (version 5.2 or greater) on your system, IC Host Recorder can take advantage of these SSL options.

Profiles

Introduction to profiles

Interaction Host Recorder is primarily a tool for building and editing host profiles, the scripts that allow handlers to send and receive data from a host session. This topic offers an overview of one profile building method, but once you are familiar with the process you can perform these steps in a different order.

Before you begin

Before you can begin creating host profiles make sure you have login and password information that the handlers will need to connect with the host.

1. Configure the connection properties

   Before you can start a host session you must configure connection properties such as host address, type, and several more. The online help for the Connection dialog explains the options. Once properly configured, you can connect to the host and open the first screen.

2. Define the screen

   After connecting, you should define the current screen. You must name the current screen before you can create validation rules, user defined fields, or transitions. Handlers cannot act upon screens that are not defined.

3. Create validation rules for the new screen

   Validation rules are the distinguishing characteristics of a specific named screen. You can define a screen by its screen ID, the appearance of certain text in a certain location, or several other characteristics. The online help for the Validation Rules list explains the different kinds of validation rules.

4. Creating user defined fields

   User defined fields are the areas on the screen where handlers can read and write information. There are eight types of user defined fields that allow you to define a set of coordinates, or a position relative to some other element on the screen. The online help for User Defined Fields explains your options. You can create user defined fields on any named screen.

5. Creating Transitions

   After defining one or more screens and setting the validation rules for those screens, you will probably want to create transitions between the screens. Transitions are the collection of instructions (transition commands) used to move from one screen to another. Interaction Host Recorder has a recording feature that allows you to automatically record transitions, or you can create and edit them manually.

6. Publishing Host Profiles

   When your host profile contains the defined screens, user defined fields, and transitions that you need, you can publish the host profile to the Host Server. The Host tools in handlers can then tell the Host server to run the profiles and perform one or more transitions within the host profile.

Host Profile Grammar (.hcs files)

This topic explains the host profile syntax. We do not recommend that you manually make changes to your host profiles; rather, you should edit your profiles using Interaction Host Recorder. This syntax is provided as an aid for advanced authoring and debugging.

Host profiles contain three sections: connection settings, screen declarations, and transitions.

Connection Settings

These settings correspond to those set up in the Connection Properties dialog .

<table>
<tr><th>Syntax</th><th>Example</th></tr>
<tr><td><connection> := CONNECTION [<name>] {<connectitems>}*</td><td>Connection LOCIS</td></tr>
<tr><td><connectitems> :=</td><td></td></tr>
<tr><td>HOSTTYPE = (Mainframe | AS400) |</td><td>HOSTTYPE = MAINFRAME</td></tr>
<tr><td>DRIVER = <string>, <string> |</td><td>DRIVER = "I3WdExpMFDriver", "I3WdExpMFDriver"</td></tr>
<tr><td>INTERFACE = (TN3270 | TN5250 | SNA3270 | SNA5250) |</td><td>INTERFACE = TN3270</td></tr>
<tr><td>ADDRESS = <string> |</td><td>ADDRESS = "locis.loc.gov"
ADDRESS = "129.71.2.10"</td></tr>
<tr><td>PORT = <posnum> |</td><td>PORT = 23</td></tr>
<tr><td>MODE = (24x80 | 32x80 | 43x80 | 27x132) |</td><td>MODE = 24x80</td></tr>
<tr><td>EXTATTR = <boolflag> |</td><td>EXTATTR = TRUE</td></tr>
<tr><td>TIMEOUT = <posnum></td><td>TIMEOUT = 30</td></tr>
<tr><td>DEVICE = <string> |</td><td></td></tr>
<tr><td>TERMINAL = <string> |</td><td></td></tr>
<tr><td>SSLENABLED = <boolflag> |</td><td>SSLENABLED = FALSE</td></tr>
<tr><td>SSLAUTHENTICATION = <boolflag> |</td><td>SSLAUTHENTICATION = FALSE</td></tr>
<tr><td>SSLLICENSINGSERVER = <string> |</td><td></td></tr>
<tr><td>LU = <string> |</td><td></td></tr>
<tr><td>LOCALLU = <string> |</td><td></td></tr>
<tr><td>PARTNERLU = <string> |</td><td></td></tr>
<tr><td>SYSTEMNAME = <string> |</td><td></td></tr>
</table>

DEVICENAME = <string> |	
AUTOSIGNON = <boolflag> |	AUTOSIGNON = TRUE
USERID = <String> |	USERID = "JohnD"
USERPASSWORD = <string>	USERPASSWORD = "MyPassword"

Screen Declarations

Screen declarations are the screens you name in the Screen Definition window. Screen declarations contain validation rules and user defined fields. The <boolflags> corespond to the option settings users encounter when editing via the Interaction Host Recorder. You can view these options creating a similar field using the appropriate dialog in Interaction Host Recorder.

Screen Declaration

Syntax	Example
<screendecl> := **SCREEN** <name> **VALIDATION(**<posnum>**) =** <validation> {, <validation>}*; {**FIELD** <name> **=** <field_decl>}*	SCREEN MainScreen

Validation Rules

Syntax	Example
<validation> :=	VALIDATION(5000) = StringAt(68, 24, "LOCI StringAt(7, 23, "Choice: 2 ", FALSE), ID(-185882193), IsReadyToSend
Wait(<posnum>**)** |	Wait(100)
ID(<number>**)** |	ID(-238147156)
Type(<number>**)** |	Type(1)
StringAt((<posnum> | ***), (<posnum> | ***), <string>, <boolflag>**)** |	StringAt(68, 24, "LOCISMENU ", FALSE)
CursorAt((<posnum> | ***), (<posnum> | ***)**)** |	CursorAt(6, 24)
IsReadyToSend	IsReadyToSend

Syntax	Example

<field_decl> :=	
AbsPos((<posnum> | *), (<posnum> | *), <length>[, <boolflag>[, <boolflag>]]) |	FIELD Field01 = AbsPos(1, 1, 80, FALSE, TRUE)
RelPos(<number>, <number>, <length>[, <boolflag>[, <boolflag>]]) |	FIELD Field02 = RelPos(52, 1, 12, FALSE, FALSE)
OrdP(<posnum>[, <boolflag>[, <boolflag>]]) |	FIELD Field03 = OrdP(1, FALSE, FALSE)
OrdU(<posnum>[, <boolflag>[, <boolflag>]]) |	FIELD Field04 = OrdU(1, FALSE, FALSE)
RelStringTopDown(<posnum>, <string>, <number>, <number>, <posnum>, <boolflag>[, <boolflag>[, <boolflag>]]) |	FIELD Field05 = RelStringTopDown(3, "USC", -6, 15, 3, FALSE, TRUE, TRUE)
RelStringBottomUp(<posnum>, <string>, <number>, <number>, <posnum>, <boolflag>[, <boolflag>[, <boolflag>]]) |	FIELD Field06 = RelStringBottomUp(3, "USC", -6, 18, 3, FALSE, TRUE, TRUE)
StringDelimitedTopDown(<posnum>, <string>, <boolflag>, <string>, <boolflag>, <number>, <number>[, <boolflag>[, <boolflag>]])	FIELD Field07 = StringDelimitedTopDown(1, "===> ", TRUE,"PF1", FALSE, 0, -44, TRUE, TRUE)
StringDelimitedBottomUp(<posnum>, <string>, <boolflag>, <string>, <boolflag>, <number>, <number>[, <boolflag>[, <boolflag>]])	FIELD Field07 = StringDelimitedBottomUp(1, "===> ", TRUE,"PF1", FALSE, 0, -44, TRUE, TRUE)

<transdecl> := **TRANSITION** (**STARTUP** | <name>) **->** <name>

COMMANDS = <transcmd> {, <transcmd>}*;

Syntax	**Example**
<transcmd> :=	TRANSITION STARTUP -> MainScreen TRANSITION CopyrightDocumentFile -> MainScreen
Validate(<name>) |	Validate(MainScreen)
PutString((<posnum> | *), (<posnum> | *), <string>) |	PutString(24, *, "hello")

PressKey(<string>) |	PressKey("1")
	PressKey("@E")

WaitForRTS(<posnum>) |	WaitForRTS(5000)
Wait(<posnum>) |	Wait(300)
WaitForString((<posnum> | *), (<posnum> | *), <string>, <boolflag>, <posnum>) |	WaitForString(6, 24, "Tolkien", FALSE, 2000)
WaitForCursor((<posnum> | *), (<posnum> | *), <posnum>) |	WaitForCursor(22, 24, 4500)
WaitForScreenID(<number>, <posnum>) |	WaitForScreenID(-238147156, 2000)

<table>
<tr><td>WaitForScreenType(<number>, <posnum>) |</td><td>WaitForScreenType(1, 2000)</td></tr>
<tr><td>SetCursor((<posnum> | *), (<posnum> | *)</td><td>SetCursor(6, 24)</td></tr>
</table>

```
<name> := (a-z|A-Z|_) {(a-z|A-Z|0-9|_)}*
<posnum> := {0-9}+
<number> := [(+|-)] {0-9}+
<string> := " {(<any_char> | <escapeseq>)}* "
<boolflag> := (TRUE | FALSE)
<netname> := {(a-z|A-Z|0-9|_|-)}*
<escapeseq> := \" | \' | \\ | \? | \a | \b | \f | \n | \r |
\t | \v | \x<hexdigit><hexdigit>
<hexdigit> := (0-9|a-f|A-F)
```

```
# Starts a line of comments (rest of the line ignored)
```

Sample Profile (.hcs file)

This topic contains the text found in a fully functional host profile. While we do not recommend that you create or edit host profiles manually, this profile serves as an sample of the types of commands you will find in a host profile. See Host Profile Grammar (.hcs files) for more information on the individual commands found in a host profile.

```
################################################

##### IC Host Connectivity Script --- Do Not Modify! #####

CONNECTION LibOfCongress

HOSTTYPE = Mainframe

DRIVER = "I3WdExpMFDriver", "I3WdExpMFDriver"

INTERFACE = TN3270

ADDRESS = "locis.loc.gov"

PORT = 23

MODE = 24x80

EXTATTR = TRUE

TIMEOUT = 30


SCREEN CommentsAndLogoff

VALIDATION(5000) = ID(-679864668),

StringAt(68, 24, "COMMENTS/OFF ", FALSE),

IsReadyToSend;


SCREEN CopyrightDocumentFile

VALIDATION(5000) = ID(7355349),

StringAt(1, 2, "***COHD- THE COPYRIGHT DOCUMENTS FILE", FALSE);


SCREEN CopyrightInfo

VALIDATION(5000) = StringAt(68, 24, "COPYRIGHT ", FALSE),

ID(-112686159);


SCREEN LocisCatalog

VALIDATION(3000) = StringAt(68, 24, "LC CATALOG", FALSE),

StringAt(*, *, "Choice", TRUE);


SCREEN MainScreen

VALIDATION(3000) = StringAt(68, 24, "LOCISMENU ", FALSE),

StringAt(*, *, "Choice", TRUE),
StringAt(7, 23, "Choice: ", FALSE),
CursorAt(16, 23),
IsReadyToSend;
FIELD Field1 = RelPos(52, 1, 12, FALSE, FALSE)
FIELD Field2 = RelStringBottomUp(1, "Choice:", 9, 0, 2, TRUE, FALSE, FALSE)
FIELD Field3 = StringDelimitedTopDown(1, "12", FALSE,"ch", TRUE, 0, 0, TRUE, TRUE)
FIELD Field4 = StringDelimitedTopDown(3, "Library", FALSE,"LOCIS", FALSE, 0, 0, FALSE,
FALSE)
FIELD Line01 = AbsPos(1, 1, 80, FALSE, TRUE)
FIELD Line02 = AbsPos(1, 2, 80, FALSE, TRUE)
FIELD Line03 = AbsPos(1, 3, 80, FALSE, TRUE)
FIELD Line04 = AbsPos(1, 4, 80, FALSE, TRUE)
FIELD Line05 = AbsPos(1, 5, 80, FALSE, TRUE)
FIELD Line06 = AbsPos(1, 6, 80, FALSE, TRUE)
FIELD Line07 = AbsPos(1, 7, 80, FALSE, TRUE)
FIELD Line08 = AbsPos(1, 8, 80, FALSE, TRUE)
FIELD Line09 = AbsPos(1, 9, 80, FALSE, TRUE)
FIELD Line10 = AbsPos(1, 10, 80, FALSE, TRUE)
FIELD Line11 = AbsPos(1, 11, 80, FALSE, TRUE)
FIELD Line12 = AbsPos(1, 12, 80, FALSE, TRUE)
FIELD Line13 = AbsPos(1, 13, 80, FALSE, TRUE)
FIELD Line14 = AbsPos(1, 14, 80, FALSE, TRUE)
FIELD Line15 = AbsPos(1, 15, 80, FALSE, TRUE)
FIELD Line16 = AbsPos(1, 16, 80, FALSE, TRUE)
FIELD Line17 = AbsPos(1, 17, 80, FALSE, TRUE)
FIELD Line18 = AbsPos(1, 18, 80, FALSE, TRUE)
FIELD Line19 = AbsPos(1, 19, 80, FALSE, TRUE)
FIELD Line20 = AbsPos(1, 20, 80, FALSE, TRUE)
FIELD Line21 = AbsPos(1, 21, 80, FALSE, TRUE)
FIELD Line22 = AbsPos(1, 22, 80, FALSE, TRUE)
FIELD Line23 = AbsPos(1, 23, 80, FALSE, TRUE)
FIELD Line24 = AbsPos(1, 24, 80, FALSE, TRUE)

SCREEN MainScreenSel1
VALIDATION(5000) = StringAt(*, *, "Choice:", FALSE),
StringAt(68, 24, "LOCISMENU ", FALSE),
StringAt(7, 23, "Choice: 1 ", FALSE);
```

```
SCREEN NoLUsAvailable
VALIDATION(3000) = StringAt(1, 4, " No LUs available now, please try later", FALSE),
ID(298566241);
FIELD UserID = OrdU(1, FALSE, FALSE)
FIELD Password = OrdU(2, FALSE, FALSE)
FIELD NewPassword = OrdU(3, FALSE, FALSE)
FIELD Application = OrdU(4, FALSE, FALSE)

SCREEN Screen1
VALIDATION(5000) = StringAt(68, 24, "LOCISMENU ", FALSE),
StringAt(7, 23, "Choice: 2 ", FALSE),
ID(-185882193),
IsReadyToSend;

SCREEN Screen10
VALIDATION(5000) = CursorAt(17, 23),
StringAt(16, 23, " ", FALSE);

SCREEN Screen2
VALIDATION(5000) = ID(0),
Type(1);

SCREEN Screen3
VALIDATION(5000) = StringAt(7, 23, "Choice: 12", FALSE);

SCREEN Screen4
VALIDATION(5000) = StringAt(7, 23, "Choice: 12", FALSE);

SCREEN Screen5
VALIDATION(5000) = CursorAt(32, 9);

SCREEN Screen6
VALIDATION(5000) = CursorAt(52, 13);

SCREEN Screen7
VALIDATION(5000) = CursorAt(52, 13);
```

```
SCREEN Screen8
VALIDATION(5000) = CursorAt(58, 17);


SCREEN Screen9
VALIDATION(5000) = StringAt(7, 19, "The Organizations (NRCM) file is no longer created
or supported by LC. It has been removed from LOCIS.", FALSE),
StringAt(7, 23, "Choice: 3 ", FALSE);


SCREEN SearchItemAfter1975
VALIDATION(3000) = StringAt(1, 2, "***LOC3- THE LIBRARY OF CONGRESS INFORMATION
SYSTEM", FALSE);
FIELD Field1 = AbsPos(27, 23, 116, FALSE, FALSE)


SCREEN SearchItems1950to1974
VALIDATION(5000) = StringAt(16, 6, "LOC2 lists books cataloged 1950-74.", FALSE);


SCREEN SearchResultPg1
VALIDATION(3000) = StringAt(70, 1, "(BKS3)", FALSE);
FIELD Subjects = StringDelimitedTopDown(1, "SUBJECTS:", FALSE,"OTHER NAMES:", FALSE,
71, 0, FALSE, TRUE)
FIELD CallNumber = RelStringTopDown(1, "LC CALL NUMBER:", 16, 0, 64, FALSE, FALSE,
TRUE)
FIELD LibEntry = StringDelimitedTopDown(1, "(BKS3)", FALSE,"LC CALL NUMBER", FALSE,
85, 0, FALSE, TRUE)


TRANSITION STARTUP -> MainScreen
COMMANDS = WaitForRTS(5000),
WaitForRTS(2000),
Validate(MainScreen);


TRANSITION CopyrightDocumentFile -> MainScreen
COMMANDS = Validate(CopyrightDocumentFile),
PressKey("exit"),
PressKey("@E"),
WaitForCursor(1, 1, 5000),
WaitForCursor(16, 23, 5000),
WaitForRTS(5000),
Validate(MainScreen);
```

```
TRANSITION CopyrightInfo -> CopyrightDocumentFile
COMMANDS = Validate(CopyrightInfo),
PressKey("3"),
PressKey("@E"),
WaitForCursor(1, 1, 5000),
WaitForCursor(27, 12, 5000),
WaitForRTS(5000),
Validate(CopyrightDocumentFile);

TRANSITION CopyrightInfo -> MainScreen
COMMANDS = Validate(CopyrightInfo),
PressKey("12"),
PressKey("@E"),
WaitForCursor(1, 1, 5000),
WaitForCursor(16, 23, 5000),
WaitForRTS(5000),
Validate(MainScreen);

TRANSITION LocisCatalog -> SearchItemAfter1975
COMMANDS = Validate(LocisCatalog),
PressKey("3@E"),
WaitForRTS(5000),
Validate(SearchItemAfter1975);

TRANSITION LocisCatalog -> MainScreen
COMMANDS = Validate(LocisCatalog),
PressKey("12"),
PressKey("@E"),
WaitForCursor(1, 1, 5000),
WaitForCursor(16, 23, 5000),
WaitForRTS(5000),
Validate(MainScreen);

TRANSITION LocisCatalog -> MainScreen
COMMANDS = Validate(LocisCatalog),
PressKey("12"),
PressKey("@E"),
WaitForCursor(16, 23, 1100),
WaitForRTS(1100),
Validate(MainScreen);

TRANSITION MainScreen -> LocisCatalog
COMMANDS = Validate(MainScreen),
PressKey("@T1@E"),
WaitForRTS(5000),
Validate(LocisCatalog);

TRANSITION MainScreen -> CopyrightInfo
COMMANDS = Validate(MainScreen),
PressKey("3"),
PressKey("@E"),
WaitForCursor(1, 1, 5000),
WaitForCursor(16, 23, 5000),
WaitForRTS(5000),
Validate(CopyrightInfo);

TRANSITION MainScreen -> SearchItemAfter1975
COMMANDS = Validate(MainScreen),
PressKey("1"),
PressKey("@E"),
WaitForCursor(1, 1, 5000),
WaitForCursor(17, 24, 5000),
WaitForRTS(5000),
PressKey("3"),
PressKey("@E"),
WaitForCursor(1, 1, 5000),
WaitForCursor(27, 23, 5000),
WaitForRTS(5000),
Validate(SearchItemAfter1975);

TRANSITION MainScreen -> MainScreen
COMMANDS = Validate(MainScreen),
PressKey("1"),
PressKey("@E"),
WaitForCursor(17, 24, 5000),
WaitForRTS(5000),
Validate(LocisCatalog),
PressKey("3"),
PressKey("@E"),
WaitForCursor(27, 23, 5000),
WaitForRTS(5000),
Validate(SearchItemAfter1975),
PressKey("booch"),
PressKey("@E"),
WaitForCursor(11, 23, 5000),
WaitForRTS(5000),
PressKey("b6"),
PressKey("@E"),
WaitForCursor(8, 19, 5000),
WaitForRTS(5000),
PressKey("exit"),
PressKey("@E"),
WaitForCursor(16, 23, 5000),
WaitForRTS(5000),
Validate(MainScreen);

TRANSITION MainScreen -> LocisCatalog
COMMANDS = Validate(MainScreen),
PressKey("1"),
PressKey("@E"),
Validate(LocisCatalog);

TRANSITION MainScreen -> Screen10
COMMANDS = Validate(MainScreen),
PressKey("1"),
PressKey("@E"),
WaitForCursor(17, 24, 1000),
WaitForRTS(1000),
PressKey("3"),
PressKey("@E"),
WaitForCursor(27, 23, 1100),
WaitForRTS(1100),
PressKey("radio"),
PressKey("@E"),
WaitForCursor(8, 18, 3600),
WaitForRTS(3700),
PressKey("2"),
PressKey("@E"),
WaitForCursor(8, 20, 1000),
WaitForRTS(1000),
PressKey("display item 2"),
PressKey("@E"),
WaitForCursor(65, 21, 1200),
WaitForRTS(1200),
PressKey("simsp"),
PressKey("@<"),
PressKey("@<"),
PressKey("@<"),
PressKey("@<"),
PressKey("@<"),
PressKey("newtons"),
PressKey("@<"),
PressKey("@E"),
WaitForCursor(8, 20, 1600),
WaitForRTS(1600),
PressKey("1"),
PressKey("@E"),
WaitForCursor(8, 20, 1000),
WaitForRTS(1000),
PressKey("exit"),
PressKey("@E"),
WaitForCursor(16, 23, 1400),
WaitForRTS(1400),
Validate(Screen10);

TRANSITION Screen9 -> LocisCatalog
COMMANDS = Validate(Screen9),
PressKey("@T"),
PressKey("1"),
PressKey("@E"),
WaitForCursor(17, 24, 1000),
WaitForRTS(1000),
Validate(LocisCatalog);

TRANSITION SearchItemAfter1975 -> MainScreen
COMMANDS = Validate(SearchItemAfter1975),
PressKey("@T"),
PressKey("exit"),
PressKey("@E"),
WaitForCursor(1, 1, 5000),
WaitForCursor(16, 23, 5000),
WaitForRTS(5000),
Validate(MainScreen);

# End of file #
```

Screens

Defining Screens

Before you can create validation rules or user defined fields on a screen, you must define that screen. While you can define a screen simply by giving it a name, Interaction Host Recorder cannot accurately identify that screen until you create validation rules for that screen.

To Define a New Screen

After connecting to the host, navigate to the screen you want to define. Once that screen is loaded, right-click in the Screen list and choose **Add Screen...** from the menu that appears. The Screen Properties dialog appears in which you can name the screen and define its validation rules . After you click OK, the screen name appears in the Screen list. Any time that screen is loaded and the validation rules match, that screen name appears in the Screen list.

After a screen is named, you can edit the screen name and validation rules by right-clicking the screen name and choosing the appropriate command from the menu that appears.

Screen Properties dialog

The Screen Properties dialog is where you name and define validation rules for a screen. This screen appears any time you choose to add a screen manually, or if you stop at an unrecognized screen when you are recording transitions automatically.

Validation Page

On this page you can view or create validation rules.

Timeout

The amount of time to wait for this screen to load before checking the validation rules.

Commands

The validation rules defined for this screen.

Fields Page

On this page you can view or create user defined fields.

Fields

The amount of time to wait for this screen to load before checking the validation rules.

Validation Rules

Validation Rules

Validation rules contain information for identifying the currently active screen. When a screen appears, Interaction Host Recorder checks the validation rules to determine which screen is loaded. If a person were sitting at the terminal emulation, that person could tell that a screen was valid just by waiting until it finished loading and reading it. When a handler is running the terminal emulation, things become more complicated. The validation rules must specify exactly what determines a valid screen. In many cases you will want to use more than one validation rule, just to make sure that the screen is correct.

When Host Server runs your profile, it will only validate the screen when instructed to do so in a transition command. This means that you will need to configure when screens are validated in your transition commands. If you are recording transition commands automatically, then the validations are inserted for you, but you can also add or remove validations manually in the Transitions Commands window.

Create and edit validation rules in the Screen Definition window

Note: For a new screen (previously undefined in this host profile), you must first name that screen before you can create validation rules.

Interaction Host Recorder evaluates each rule in your list, starting with the top rule in the list. Once it has determined that the first rule is valid, it goes on to the next rule in the list. Interaction Host Recorder will continue to evaluate a rule until it succeeds or times out. (You can set the time out value by right-clicking in the Screen Definition window and choosing Screen Properties from the dialog that appears.

There are six types of validation rules:

- **Screen ID**
  Each screen has a unique identifier generated mathematically from the number, position, and length of the fields. If you know that the fields will always be in the same place and of the same length, you can use this value. You can use screen ID even if some of the values within fields are going to change. For example, if there is a time stamp located in the top left corner, and that field is always the same length, the screen will keep it's unique id even if the time of day is different.

- **Screen Type**
  The number of unprotected fields on the screen. If you know that a screen has finished loading when an input field has appeared, use the screen type as one of your validation rules. It is possible that several different screens will have the same number of input fields, and thus have the same screen type. In most cases, screen type should not be used as the only validation rule. This validation rule should only be used when a screen will always have the same number of unprotect6ed fields.

- **Match Text**
  This versatile validation rule allows you to look for a specific text string anywhere on the screen, anywhere starting at a specific row, anywhere starting in a specific column, or starting in a specific row/column coordinate. You should use this validation rule when you know that each screen has a unique identifier somewhere on the screen. Many mainframe application designers realize that their screens will be evaluated electronically, so they include a unique number in a corner of the screen.

- **Wait**
  Forces a pause while Interaction Host Recorder validates the screen. Wait should only be used when no other validation rules can be used to validate the screen.

  Caution: Be careful that you don't set the wait time for longer than the validation timeout or the validation will always fail.

- **Cursor Position**
  Looks for the cursor in a specific row, column, or row/column coordinate. For example, you might use cursor position when you know that the cursor will be in row twenty-five when the screen has finished loading. You should not use cursor position if the cursor is not always in the same location after a screen loads.
- **Is Ready to Send (RTS)**
  Checks to see if the host has sent an RTS signal indicating that the screen is ready to accept user input. You might use this as a final validation rule after other rules have been checked. For example, if the host sends between fifteen and twenty RTS signals before the screen finishes loading, then you might Match Text for a textual value at the bottom of the screen and then validate the RTS signal. This way you know that the screen has finished loading and is ready for input.

Creating Validation Rules

Validation rules contain information for identifying the currently loaded screen. When a screen appears, Interaction Host Recorder checks the validation rules to determine which screen is loaded. While you can configure Interaction Host Recorder to validate a screen every time you make a change on that screen (see Preferences), the handlers will only validate a screen when instructed to do so by a transition command.

Before you can create a validation rule, you must connect to a host and define the screen for which you will create validation rules. Once you have defined the screen, make sure that screen is loaded and selected in the Screen Definition window create a validation rule by right-clicking in the Validation Rules list and choose **Add Validation...** from the menu that appears. The Validation dialog appears in which you can configure the validation rule type and settings. See Validation Rules for more information on the types of validation rules.

Once you click OK, the validation rule appears in the Validation Rules list. You can edit the properties of this validation rule at any time by right-clicking the validation rule and choosing **Edit Validation...** from the menu that appears.

User Defined Fields

User Defined Fields

User defined fields are areas you define in a screen to allow handlers to write information to and from that screen. In short, they allow handlers to interact with the screens. The Host tools in handlers can bind variables in the handlers to user defined fields in the host profile. This allows handlers to write information to and from the screen.

User defined fields are listed in the Screen Definition window.

Note: You must create user defined fields, as opposed to screen fields that Interaction Host Recorder creates automatically when a screen is opened. Create, edit, and delete user defined fields by right-clicking in the User Defined Fields area and choosing the appropriate command from the menu that appears.

The following list contains links to detailed descriptions of the types of user defined fields that you can create:

- **Absolute Position**
  Absolute position defines a field at a specific set of coordinates. Typically these are useful when you can accurately predict the contents of a screen, such as a menu screen.
- **Relative Position**
  Relative position uses the current cursor position as a starting point. While you can perform the same operation with Absolute coordinates, Relative Position allows you to offset the field from the cursor. This is useful if you cannot predict the contents of the screen, but you know that the cursor is always at the input field when the screen has loaded.
- **Ordinal**
  Ordinal protected and unprotected target a specific field in a series of fields. For example, if you know that a screen always has three unprotected fields, you could specify the first, second, or third unprotected field as a user defined field.
- **Anchor String**
  Anchor strings allow you to define a field based on the location of a certain word or character in the screen.
- **String Delimited**
  String delimited user defined fields are bounded by two strings that you specify. This method is useful when you don't know how long the string will be, but you can define a textual start and end point.

Creating and Editing User Defined Fields

User defined fields are areas you define in a screen to allow handlers to write information to and from that screen. In short, they allow handlers to interact with the screens. The Host tools in handlers can bind variables in the handlers to user defined fields in the host profile. This allows handlers to write information to and from the screen.

Creating User Defined Fields

After you have defined a screen, you can create user defined fields within that screen. (We recommend that you create validation rules when you define a screen.)

Use your mouse pointer to select an area on the screen, then right-click in the selected area and choose **Add User Defined Field...** from the menu that appears. The User Defined Field dialog appears. Name your field and select the type of field you wish to add. For an explanation of the types of fields you can define, see the online help in the User Defined Field dialog. When you click the OK button, the new user defined field appears in the User Defined Fields list.

Editing User Defined Fields

You can edit the properties of a user defined field by right-clicking that field in the User Defined Field list and choosing **Edit Field...** from the menu that appears. The User Defined Field dialog opens where you can make your changes.

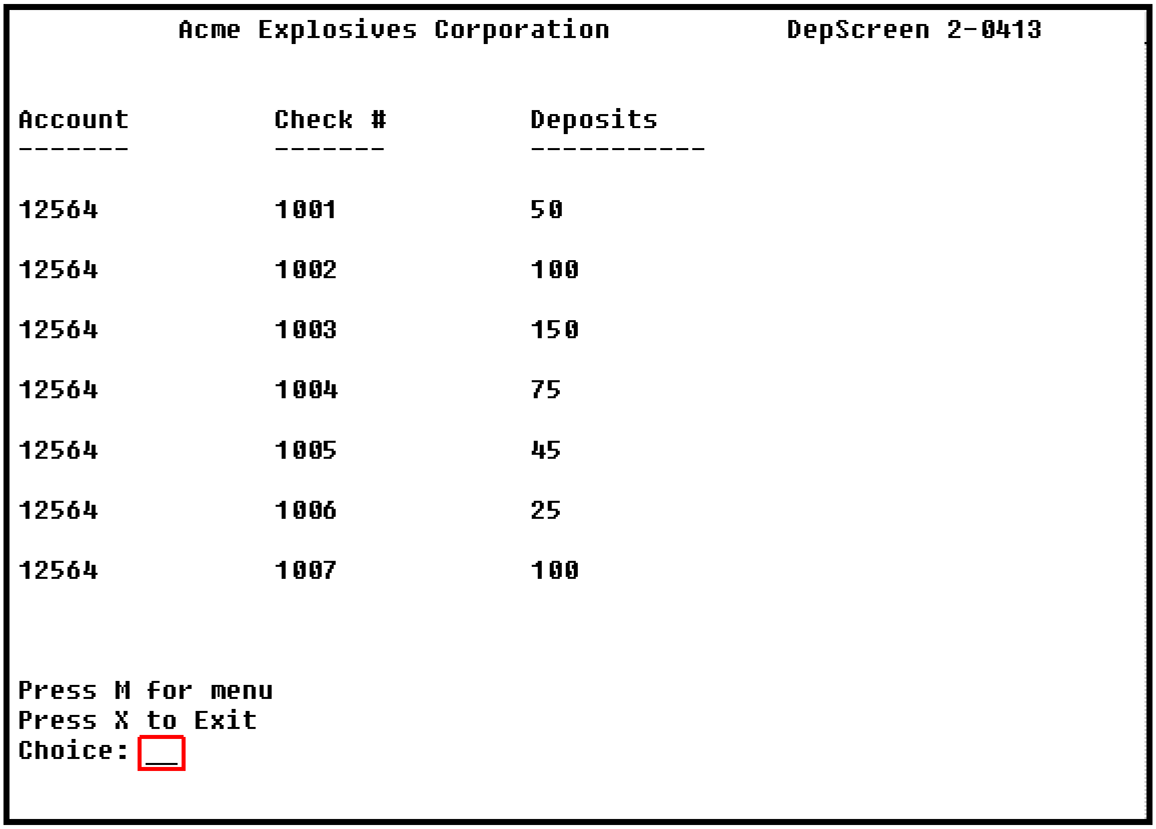
Types of User Defined Fields

Absolute Position User Defined Fields

Using absolute position is really just assigning the screen coordinates where the field will be located. Absolute coordinates allow you to define a specific column location and row location on a screen. You can also define row or column (or both) at the current cursor position. Using cursor position introduces flexibility because handlers can move the cursor on the screen, moving the field with the cursor. This topic describes the four ways in which you can use absolute position to define a field.

- **Specify row and column coordinates** to define an exact position on the screen. Use this method if you are sure that the value must be retrieved from a precise row and column coordinate every time.

  Example: Absolute field at Row=23, Col=9, length=2

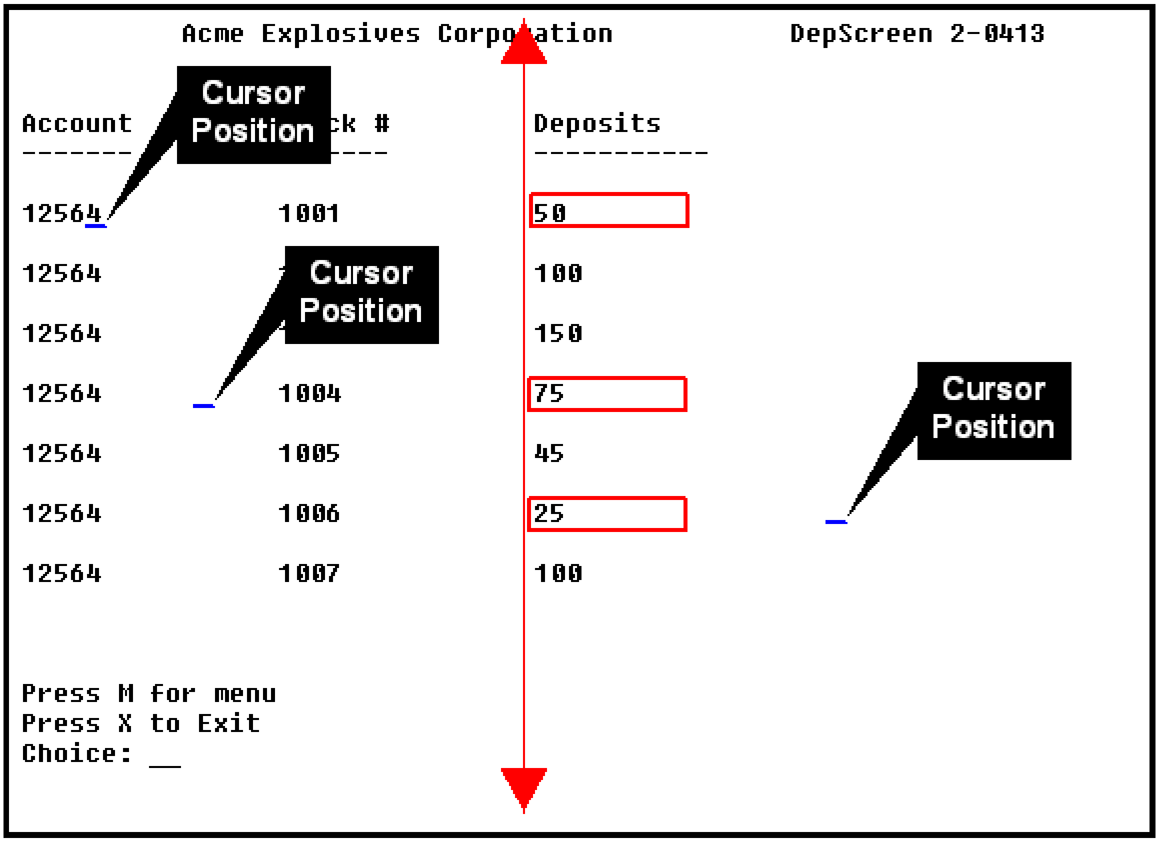


```
          Acme Explosives Corporation          DepScreen 2-0413


Account          Check #          Deposits
-------          -------          -----------

12564            1001             50

12564            1002             100

12564            1003             150

12564            1004             75

12564            1005             45

12564            1006             25

12564            1007             100



Press M for menu
Press X to Exit
Choice: __
```

- **Specify only a column position** if you know that the field will always be in a specific column. No matter which row the cursor is in, the field will be in the specified column on the cursor row. The Host Move to Position tool in a handler can move the cursor down the column.

  This method is useful with tabular data is in a vertical column, and you want to retrieve data down that column.

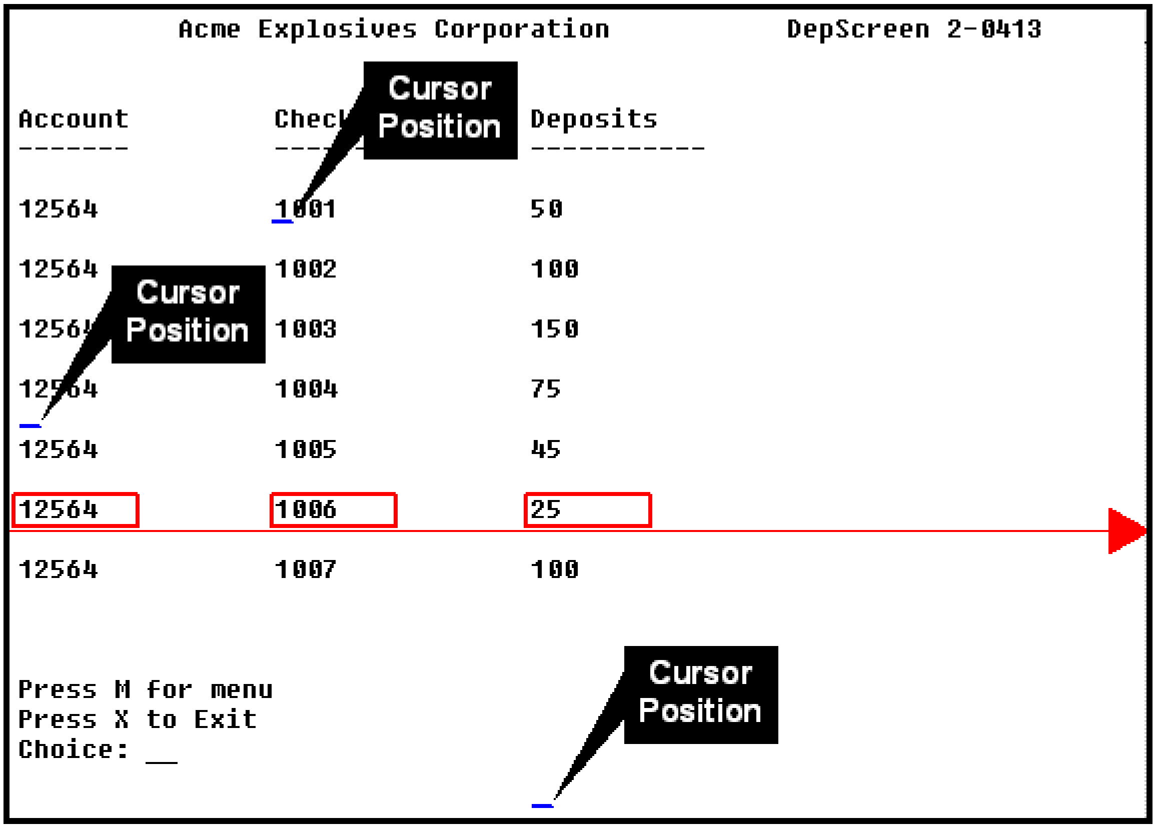
  Example: Absolute field at Row=Cursor, Col=31, length=9



```
           Acme Explosives Corporation          DepScreen 2-0413

Account            ck #          Deposits
--------           ----          -----------

12564              1001          50

12564                            100

12564                            150

12564              1004          75

12564              1005          45

12564              1006          25

12564              1007          100


Press M for menu
Press X to Exit
Choice: __
```

- **Specify only a row position** if you know the field will be in specific row. No matter which column the cursor is in, the field will be in the specified row. This method is useful with tabular data is in a horizontal row, and you want to retrieve data across that row. The Host Move to Position tool in a handler can move the cursor across the row.

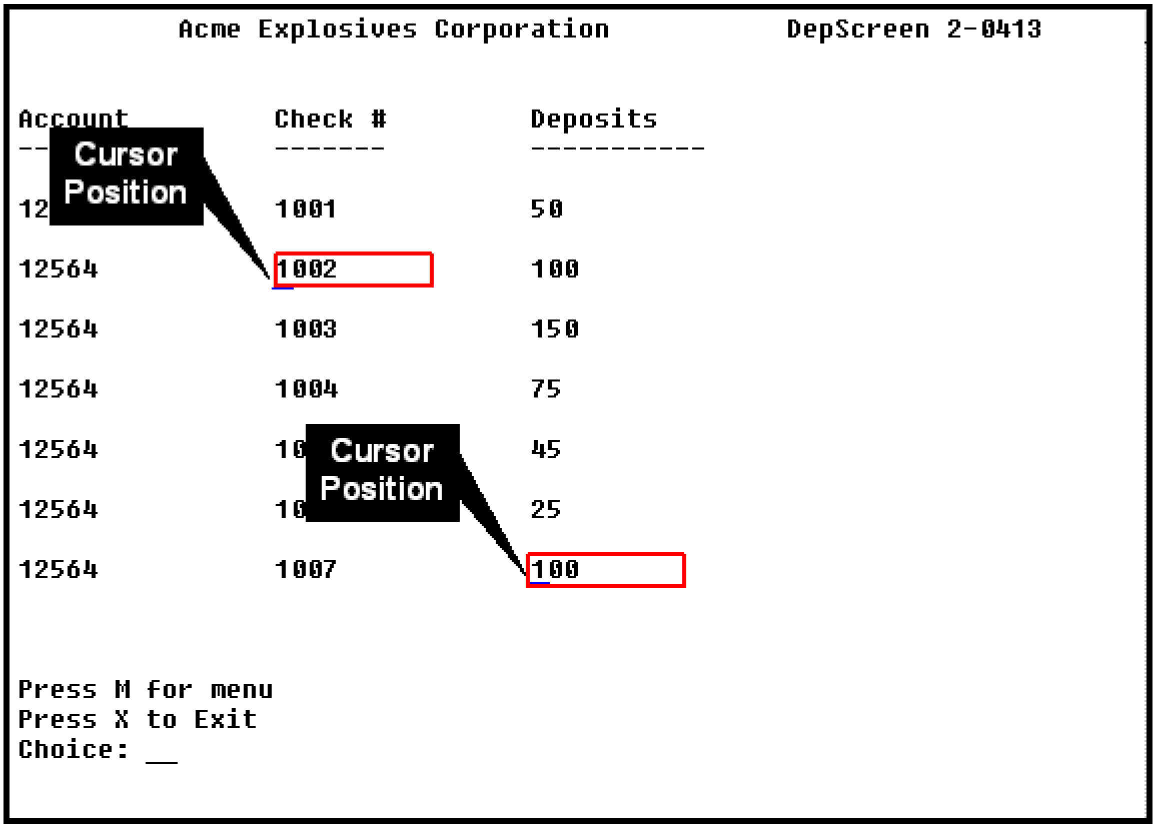
Example: Absolute field at Row=12, Col=Cursor, length=7



```
          Acme Explosives Corporation          DepScreen 2-0413

Account          Chec              Deposits
--------         ---               -----------

12564            1001              50

12564            1002              100

1256             1003              150

12564            1004              75

12564            1005              45

12564            1006              25

12564            1007              100


Press M for menu
Press X to Exit
Choice: __
```

- **Set Row and Column to Cursor Position** if you know that the cursor always ends up at a specific location. Using this method, the field starts at the current cursor position, wherever that is on the screen. If you want the field to start before or after the current cursor position, use relative position because it allows offsetting.

  Example: Absolute field at Row=Cursor, Col=Cursor, length=7



```
           Acme Explosives Corporation           DepScreen 2-0413


Account                 Check #                 Deposits
--                      -------                 -----------

12                      1001                    50

12564                   1002                    100

12564                   1003                    150

12564                   1004                    75

12564                   10                      45

12564                   10                      25

12564                   1007                    100



Press M for menu
Press X to Exit
Choice: __
```

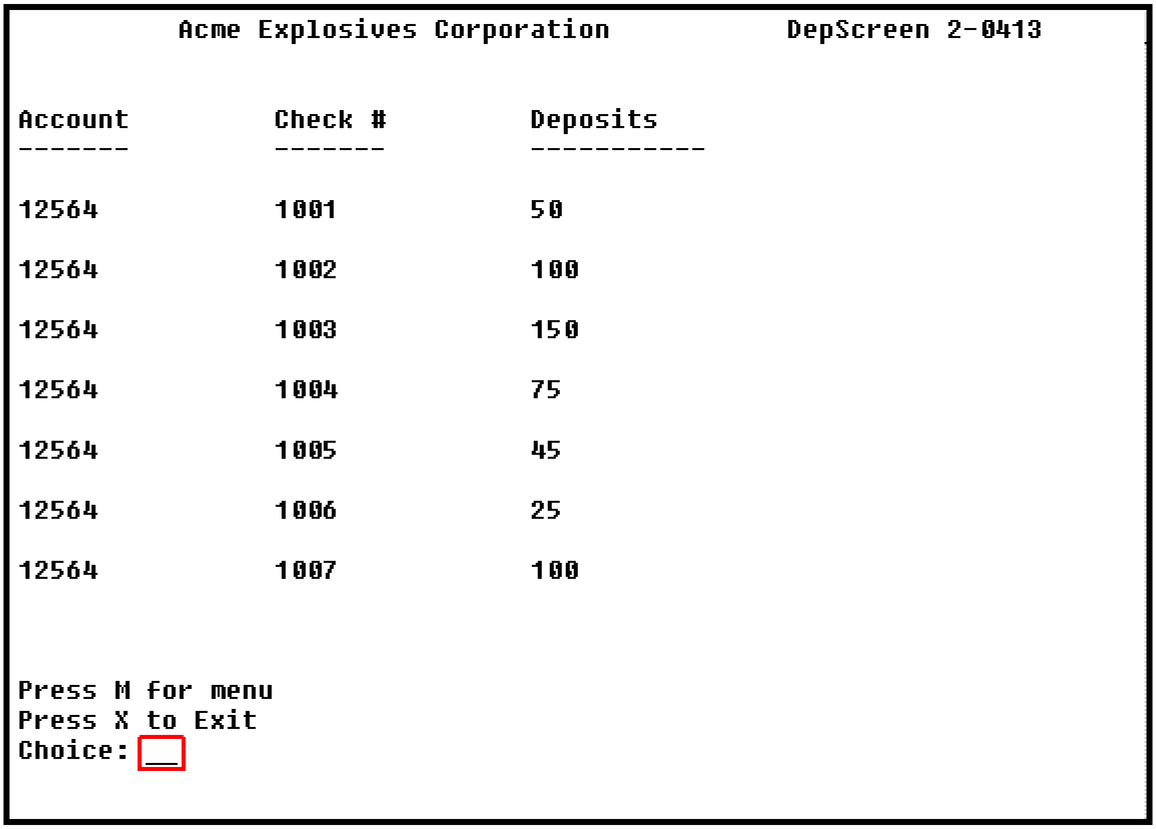
Anchor String User Defined Fields

Anchor strings allow you to define a field based on the location of a certain word or character in the screen. For example, if you know that the word "Choice:" appears on a screen, you can define a field based on the location of Choice.

When defining anchor strings, you can specify anchor string from top, or anchor string from bottom. Choose top if you want to start searching for occurrences from the top of the string; choose bottom if you want to start searching for occurrences from the bottom of the screen.

Use anchor string when you can't predict the contents of a screen, such as tabular data of undetermined length, or when you can't predict the number of protected or unprotected screen fields. In short, anchor string provides an alternative method when other methods are not reliable.

Example: Occurance = 1, Text = Choice:, Row offset = 0, Column offset = 9



```
          Acme Explosives Corporation           DepScreen 2-0413


Account         Check #         Deposits
-------         -------         -----------

12564           1001            50

12564           1002            100

12564           1003            150

12564           1004            75

12564           1005            45

12564           1006            25

12564           1007            100



Press M for menu
Press X to Exit
Choice: __
```

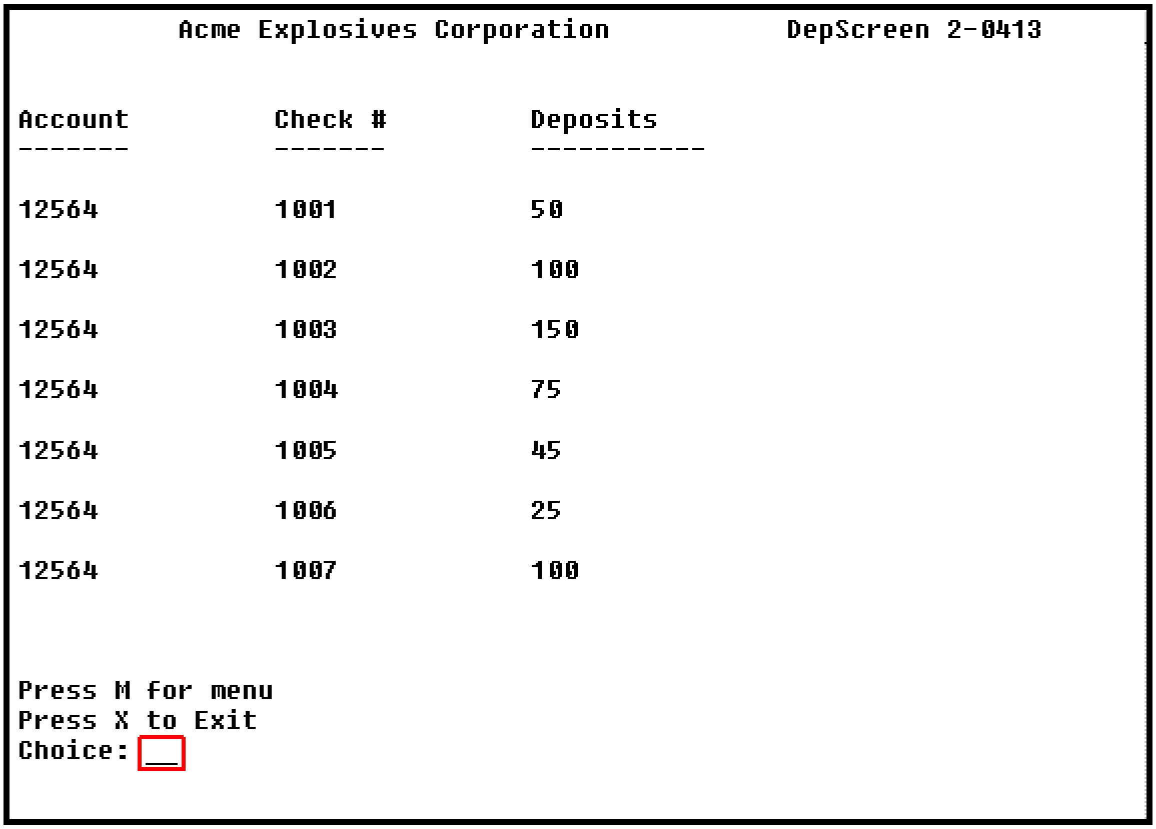
Ordinal User Defined Fields

Ordinal definitions allow you to define a specific field in a series of fields. When mainframe programmers design screens, they may specify the number of protected or unprotected fields in that screen. For example, a menu may contain 10 protected fields and 1 unprotected field. You can use this to your advantage by creating a user defined field at the first unprotected field. In this example, your user defined field would always be at the correct location for specifying the menu entry.

You should use ordinal field definitions if the number of fields (either protected or unprotected) on a screen is constant.

See Screen Fields for more information on protected and unprotected screen fields.

Example: Ordinal Unprotected Ordinality 1



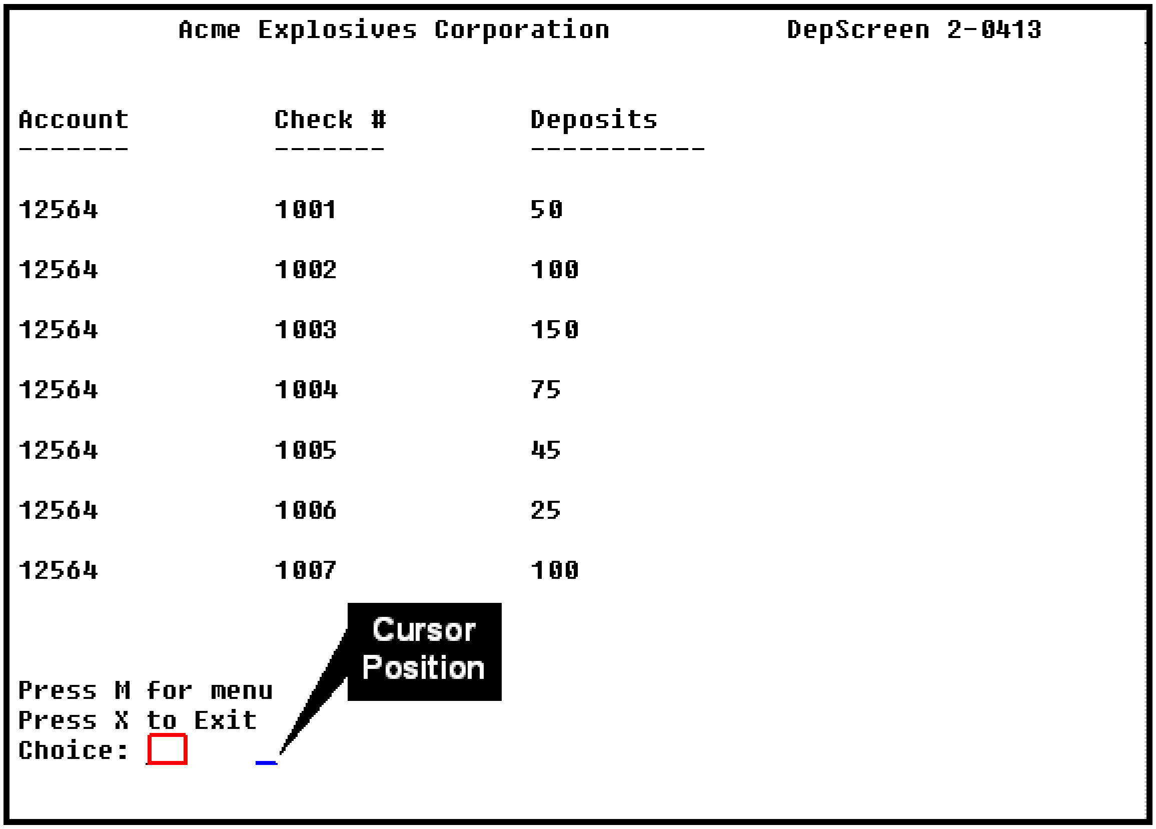
```
          Acme Explosives Corporation           DepScreen 2-0413


Account          Check #          Deposits
-------          -------          -----------

12564            1001             50

12564            1002             100

12564            1003             150

12564            1004             75

12564            1005             45

12564            1006             25

12564            1007             100



Press M for menu
Press X to Exit
Choice: __
```

Relative Position User Defined Fields

Relative position uses the current cursor position as a reference point. While you can perform the same operation with absolute position, relative position allows you to offset the field from the cursor. You can offset by rows and columns, and by positive or negative values. This allows you to specify a field above, below, before, or behind the current cursor position.

Use relative position if you can reliably predict the cursor location. One use for this method is for menu entries, such as when the cursor is located in the choice field.

Example: Row offset = 0, Column offset = -7, Length = 2

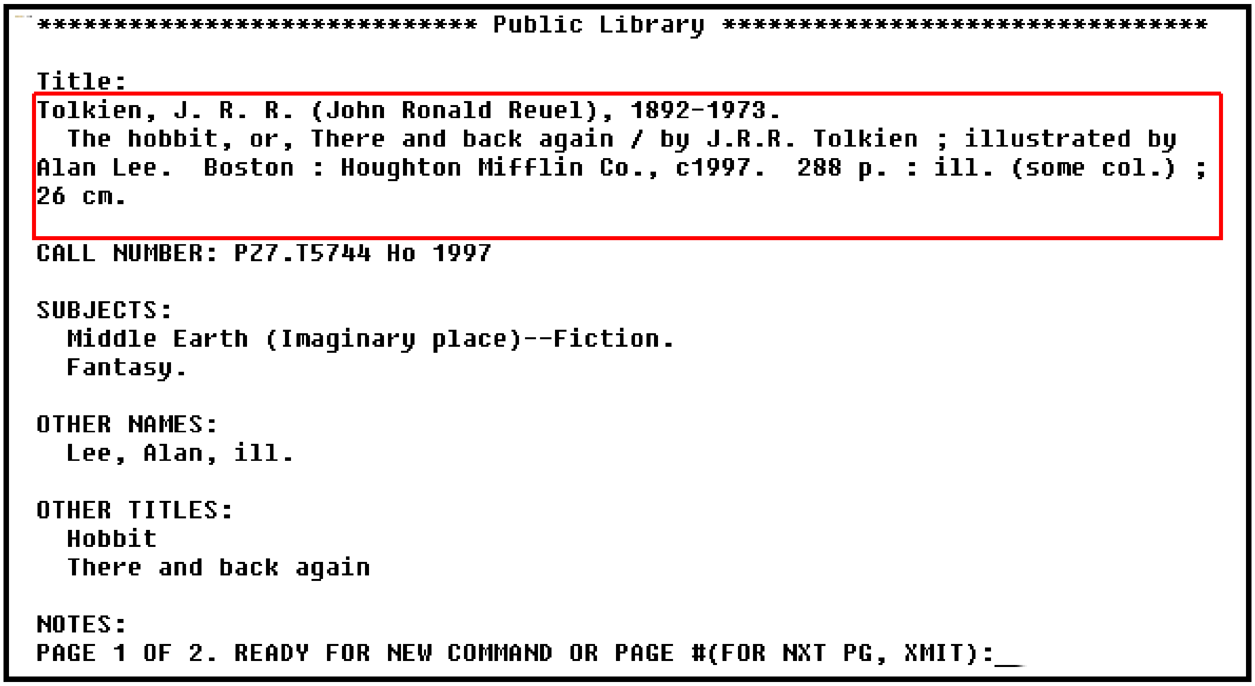


```
           Acme Explosives Corporation           DepScreen 2-0413


Account          Check #          Deposits
-------          -------          -----------

12564            1001             50

12564            1002             100

12564            1003             150

12564            1004             75

12564            1005             45

12564            1006             25

12564            1007             100


Press M for menu
Press X to Exit
Choice:
```

String Delimited User Defined Fields

String delimited user defined fields are bounded by two strings that you specify. This method is useful when you don't know how long the string will be, but you can define a start and end point. In the following example, a field is defined between the word "Title:" and the word "Call Number". As you can see in the example, this field could change length depending upon the number of characters in the field.

Example: Left Delimiter = Title, Right Delimiter = Call Number:, Occurrence = 1, Left Field Offset = 71.



```
***************************** Public Library ********************************

Title:
Tolkien, J. R. R. (John Ronald Reuel), 1892-1973.
  The hobbit, or, There and back again / by J.R.R. Tolkien ; illustrated by
Alan Lee.  Boston : Houghton Mifflin Co., c1997.  288 p. : ill. (some col.) ;
26 cm.

CALL NUMBER: PZ7.T5744 Ho 1997

SUBJECTS:
  Middle Earth (Imaginary place)--Fiction.
  Fantasy.

OTHER NAMES:
  Lee, Alan, ill.

OTHER TITLES:
  Hobbit
  There and back again

NOTES:
PAGE 1 OF 2. READY FOR NEW COMMAND OR PAGE #(FOR NXT PG, XMIT):_
```

Transitions

Transition Commands

Transition commands are the actions taken to move from one screen to another. Transitions are made up of one or more transition commands. Transition commands are listed in the Transition Commands window, and they are executed in the order they are listed.

To edit a transition command, right-click on the transition command and choose **Edit...** from the menu that appears.

Note: Some transition commands are not recorded automatically, and you must insert them manually after you record a transition. Transition commands colored **red** in the list below are recorded automatically. To manually insert a transition command, right-click in the transition command list and choose **Insert Command...** from the menu that appears. You can then move the transition command up and down in the list by right-clicking the transition command and choosing **Move Up** or **Move Down** from the menu that appears.

There are ten types of transition commands:

- **Validate**

  Validates the current screen using the validation rules prescribed for that screen.
- **Press Key**

  Sends one or more keys to the screen. You can remap your keyboard and map multiple keys to a single key press in the Preferences dialog.
- **Wait for RTS**

  Instructs the transition to wait for a ready to send signal before executing the next transition command. Use this transition command carefully because you may not be able to predict the number of ready to send signals that the host will send while loading a screen. As a result, the screen may not fully loaded before the next transition command is executed.
- **Wait for String**

  Instructs the transition to wait for the appearance of a specific string on the host screen. You can configure which string to watch for, where the screen should appear, and several other properties. This transition command must be inserted manually.
- **Wait for Cursor**

  Instructs the transition to wait for the appearance of the cursor on the screen. You can optionally specify a cursor location, and the amount of time to wait before timing out.
- **Wait for Screen ID**

  Instructs the transition to wait for a specific screen ID. This transition command must be inserted manually.
- **Wait for ScreenType**

  Instructs the transition to wait for a specific screen type . You can optionally specify a timeout value. This transition command must be inserted manually.
- **SetCursor**

  Sets the cursor at a specific row and column on the screen. You can optionally choose to set the cursor in the current row, column, or both.

- **PutString**

  Writes a specific string to a specific row and column on the screen. You can optionally choose to write the string in the current row, column, or both. This transition command must be inserted manually.

- **Wait**

  Instructs the transition to wait for a specific number of seconds before executing the next transition command. In short, this inserts a pause in the execution of transition commands.

Creating Transitions

Transitions contain the transition commands that Host Server uses to navigate between screens. There are two ways to create transitions. The first way is to begin recording and create the transitions and transition commands automatically, naming screens as you go. The second way is to first name all of the screens through which you will navigate and then create the transitions manually. Both methods are discussed below.

Creating Transitions and Transition Commands Automatically

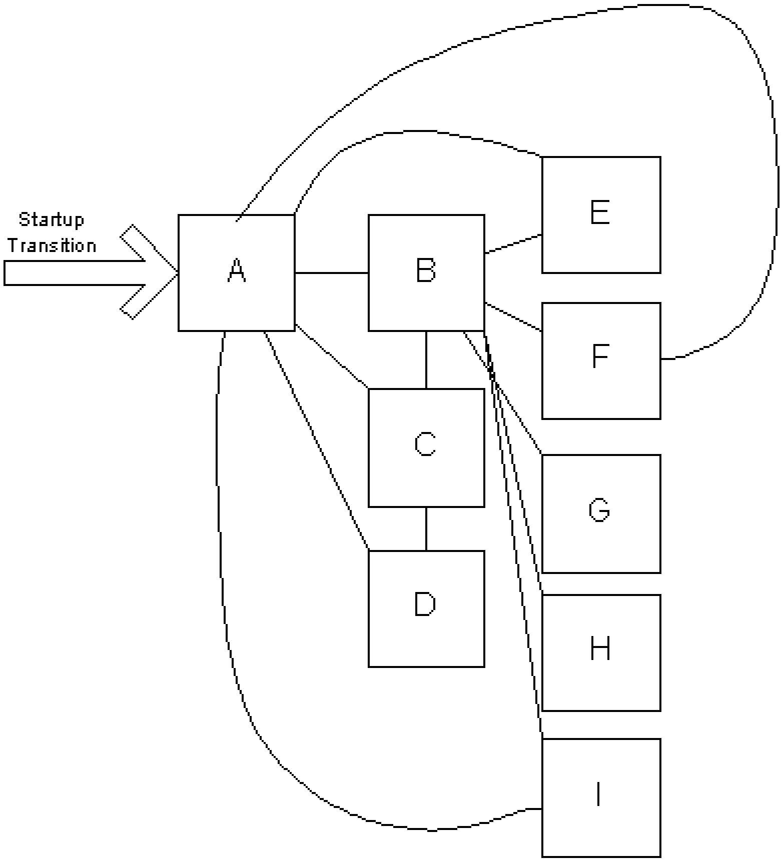
While Interaction Host Recorder has many features, automatically recording transitions is one of the most powerful. The recording feature allows you to record transitions, naming screens and creating transition commands as you go. Once you have recorded one or more transitions, you can go back and edit the screen validation rules and transition commands.

- **Step 1: Choosing a starting screen**
  If you are recording a normal transition from screen A to screen B, you must first decide which screen will be the starting point. With a normal transition, you should start with a defined (named) screen. Once you have navigated to the starting screen, start the recorder by clicking the Record button.

  If you are recording the start up transition, you should start recording before connecting to the host.

- **Step 2: Generating the Transition Commands**
  With the recorder running, navigate though the screens to the desired destination. Notice that the new transition commands are listed in the transition command window. Each time you do something on the screen, a new transition command appears in the list. (You can edit these when you finish recording.)

  Transitions can cross multiple screens. The following diagram represents a collection of related screens. If you were recording transitions, you could create a startup transition and transitions from A to B, B to H, A to F, E to A, or any combination.



- **Step 3: Defining the Target Screen**
  When you have arrived at the destination screen and have finished working in that screen, press the Stop Recording button. If the target screen is recognized, the Transition Target Screen dialog appears. If the target screen is not recognized, the Screen Properties dialog appears where you can name the screen and define validation rules. Once you have selected (or created) a name for the target screen, the transition appears as the currently selected transition list.
- **Step 4: Testing and Debugging the Transition**
  Now that you have created the transition, you should review the list of transition commands that were created automatically and make any necessary changes. The transition commands for the selected transition appear in the Transition Commands window. See transition commands for detailed information on each of the ten types of transition commands. See debugging transtion for more information on cleaning up the auto-generated transition commands.

Creating Transitions Manually

While recording transitions automatically is almost always the best method, you can also create a transition manually. You do not have to be connected to the host when creating transitions, but you must have the profile open in Interaction Host Recorder. To create a transition command manually, right-click the transition list in the transition commands window and choose **Add Transition...** from the menu that appears. Once you choose a Transition Target, you can manually insert and configure transition commands.

When you have finished, debug the transition.

Debugging Transitions

Whether you recorded your transition automatically or created a one manually, testing and debugging the transition is important before you publish a host profile to the Host Server.

Two buttons in the Interaction Host Recorder tool bar allow you to replay transitions, either all at once or one transition command at a time.

▶∞	Replays all the transition commands listed in the Transition Commands window. **Note:** Remember that the transition commands are for navigating from screen A to screen B. You must have screen A loaded before you replay the transition commands or they may fail. In fact, you should define validation rules and transitions so that they will fail if started from the wrong screen. This way you can manage the errors from within handlers.
▶1	Replays a single transition command for the currently loaded host screen. **Note:** Remember that the transition commands are for navigating from screen A to screen B. You must have screen A loaded before you replay the transition command or it may fail.

If you see that one of the transition commands is causing problems, you can edit that transition command by right-clicking it and choosing **Edit...** from the menu that appears. See Transition Commands for more information on specific transition commands.

Transition Commands window

The Transitions Commands window is where you create and edit transitions and the transition commands they contain. Transitions are stored in the host profile, and when the Move to Screen tool requests a screen move, Host Server looks in the host profile for the transition commands.

Transition Target Screen dialog

This dialog lists all the defined screens that match the current screen. When you finish recording a transition, this dialog appears for you to select the target screen.

If there are no matching screens listed or if you don't want to use any of the matching screens, you can define a new screen by clicking the **New...** button, which causes the Screen Properties dialog to appear.

Trace History window

The Trace History window is a log of all commands sent from the Interaction Host Recorder to the Host and the Host's responses. This includes all cursor movements, key presses, and screen changes. You can use the information in this window for debugging. For example, if a transition playback fails for unknown reasons, check the Trace History window to determine which commands failed to reach the host.

The information contained in this window accumulates as long as you are running Interaction Host Recorder. If you load a new host profile, the new commands are appended to the trace history log. Click the Clear button to remove all entries from the Trace History window.

The trace history related command line parameters are listed below:

```
\trace={type}

Sets level of tracing added to the trace buffer and shown in the trace history dialog.

{type}:

none - No tracing

error - Trace only errors

warning - Trace errors and warnings

status - Default, trace errors, warnings and status information
```

Notes: All of the above include additional verbose information about execution process.

The parameter keywords are not case sensitive.

Screen Fields

Host Screen Fields window

All host screens are made up of protected and unprotected fields. When you open a screen in Interaction Host Recorder, all the screen fields are listed in this window. You can use screen fields to define validation rules and user defined fields.

- **Protected screen fields**
  These are read-only fields in which data is displayed. Handlers cannot change the value in a protected screen field. Examples of a protected field include a screen label, a screen ID number, or account information in tabular form.

- **Unprotected screen fields**
  These are read/write fields into which handlers can write data. An example of an unprotected field is a Choice where you type a menu entry to change screens.

Creating validation rules from screen fields

Right-click on a screen field and choose **Add as Validation Rule...** to add the selected field as a validation rule. Interaction Host Recorder creates a Match Text type validation rule. Right-click on the new validation rule if you want to make changes.

Creating user defined fields from screen fields

Right-click on a screen field and choose **Add as User Defined Field** to add the selected field. You can define the field as absolute or ordinal.

Screen Fields (definition)

Screen fields are the fields that Interaction Host Recorder can read when a screen is opened. They are defined by the person who programmed the screens, and Interaction Host Recorder lists them in the Host Screen Fields window.

Screen fields can be protected or unprotected:

- **Protected screen fields**
  These are read-only fields in which data is displayed. If you create a user defined field from a protected field, a handler can only read the data. You cannot change the value in a protected screen field. Think of protected fields as fields that can be read but not changed.
- **Unprotected screen fields**
  These are read/write fields into which data can be written. If you create a user defined field from an unprotected screen field, a handler can write data to the screen. An example of an unprotected field is a Choice where you type a menu entry to change screens.

See Host Screen Fields window for more information on screen fields.

Configuration Options

Mnemonic keys (definition)

Keyboard mnemonics are ASCII characters that represent non-data keystrokes such as special function keys. For example, by default the down-arrow key on your keyboard sends a @V command to the host screen. In a transition command, if only one mnemonic key is sent to the host screen, Interaction Host Recorder will attempt to identify the label associated with that mnemonic. For example, if you press the down arrow key once in a command, Interaction Host Recorder displays the text "Cursor Down" in the transition command. If you press the down arrow key twice in one command, @V@V appears. This is due to a limitation in Interaction Host Recorder's ability to parse complex mnemonic sequences.

The following list shows all the mnemonic keys mapped my default, You can change the mnemonic values and their labels on the Keyboard Map page in the Preferences dialog.

Mnemonic Mappings

Meaning	Mnemonic	Mainframe	AS/400
@	@@	X	X
Alt	@A	X	
Alt Cursor	@e@1	X	X
Attention	@A@Q	X	X
Backspace	@<	X	X
Backtab (Left Tab)	@B	X	X
Clear	@C	X	X
Cmd (function) Key	@A@Y		X
Cursor Down	@V	X	X
Cursor Left	@L	X	X
Cursor Right	@Z	X	X
Cursor Select	@e@1	X	
Cursor Up	@U	X	X
Delete	@D	X	X
Dup	@S@x	X	X
End	@q		X
Enter	@E	X	X
Erase EOF	@F	X	X
Erase EOL	@e@d		X
Erase Input	@A@F	X	X
Field Exit	@A@E		X
Field Mark	@S@y	X	
Field -	@A@-		X
Field +	@A@+		X
Help	@H		X

Hexadecimal	@A@X		X
Home	@0 (zero)	X	X
Insert	@I	X	X
Insert Toggle	@A@I		X
Local Help	@e@l		X
Local Print	@P		X
Local Help Cursor	@e@m	X	X
Left Tab (Back Tab)	@B	X	X
New Line	@e@n	X	X
Page Up	@u		X
Page Down	@v		X
Print (PC)	@A@t	X	X
Print Screen	@A@T	X	
Record Backspace	@A@<		X
Reset	@R	X	X
Right Tab (Tab)	@T	X	X
Shift	@S	X	
Sys Request	@A@H	X	X
Tab (Right Tab)	@T	X	X
Test	@A@C		X
PA1	@x	X	
PA2	@y	X	
PA3	@z	X	
PA4	@+	X	
PA5	@%	X	
PA6	@&	X	
PA7	@'	X	
PA8	@(	X	
PA9	@)	X	
PA10	@*	X	
PF1/F1	@1	X	X
PF2/F2	@2	X	X
PF3/F3	@3	X	X
PF4/F4	@4	X	X
PF5/F5	@5	X	X
PF6/F6	@6	X	X
PF7/F7	@7	X	X

PF8/F8	@8	X	X
PF9/F9	@9	X	X
PF10/F10	@a	X	X
PF11/F11	@b	X	X
PF12/F12	@c	X	X
PF13/F13	@d	X	X
PF14/F14	@e	X	X
PF15/F15	@f	X	X
PF16/F16	@g	X	X
PF17/F17	@h	X	X
PF18/F18	@i	X	X
PF19/F19	@j	X	X
PF20/F20	@k	X	X
PF21/F21	@l	X	X
PF22/F22	@m	X	X
PF23/F23	@n	X	X
PF24/F24	@o	X	X

General Preferences

Preferences (General page)

The Preferences dialog is where you can change how Interaction Host Recorder behaves. On the General page you can configure several settings that will effect all of your editing sessions. These settings only affect how Interaction Host Recorder behaves while you are editing, and they do not change any values within the host profiles.

Validation timeout

The amount of time Interaction Host Recorder will wait for a validation rule to be evaluated before returning a value of false. Increase this value if you want Interaction Host Recorder to wait longer for screens to validate.

Transition commands timeout

The amount of time Interaction Host Recorder will wait for each transition command to execute before failing.

Track Current Screen option

Select this option if you want Interaction Host Recorder to validate and determine the name of the currently loaded screen each time a change is made to that screen. If this option is not selected, Interaction Host Recorder will not check to see if the current screen is a screen already defined in your host profile.

Show screen field parameter option

Select this option to view additional details about the screen fields.

Undo Stack Depth

Sets the number of undo levels available.

Other pages in the Preferences dialog:

- Preferences (Appearance page)
- Preferences (Host Fields page)
- Preferences (Keyboard Map page)

Preferences (Host Fields page)

The Preferences dialog is where you can change how Interaction Host Recorder behaves. On the Host Fields page you can configure the appearance of the unprotected and hidden screen fields. These settings only affect how Interaction Host Recorder behaves while you are editing, and they do not change any values within the host profiles.

Unprotected Fields Text Color

The color of the text that appears in the unprotected fields. This setting allows you to override the color chosen by the screen's designer. Change this setting to emphasize unprotected fields. Select automatic to disable this override.

Unprotected Fields Background Color

The background color of the unprotected fields. This setting allows you to override the color chosen by the screen's designer. Change this setting to emphasize unprotected fields. Select automatic to disable this override.

Unprotected Fields fill characters

The character to display instead of blank spaces in an unprotected field.

Hidden Field options

Select Not Displayed to hide the contents of hidden fields. Select Fill with character to obfuscate any characters with a character you select. Select Show Contents to not obfuscate the contents of hidden fields. This selection is important if you want to obfuscate password entries.

Field Color Map

In this area you can re-map any colors programmed into the main screen.

Other pages in the Preferences dialog:

- Preferences (General page)
- Preferences (Appearance page)
- Preferences (Keyboard Map page)

Preferences (Appearance page)

The Preferences dialog is where you can change how Interaction Host Recorder behaves. On the Appearance page you can configure several settings that will affect the colors and fonts of the host screen. These settings only affect how Interaction Host Recorder behaves while you are editing, and they do not change any values within the host profiles.

Border color

The color of the border around the screen.

Background color

The color of the screen background.

Host field outline color

The color of the border around the Host Screen fields when those fields are selected in the Host Screen Fields window.

Screen outline color

The color of the thin outline between the screen and the border.

Selection color

The color of the border around a manually selected portion of the screen.

Note: The color you specify is compared with a logical XOR with the color on the screen . This ensures that the selection is always visible against the background. Do not specify black as the selection color.

Bound field outline color

The color of the border around user defined fields when those fields are selected in the User Defined Fields list.

Font name

The name of the font used when displaying text in a screen.

Preserve aspect ratio of font while in host screen view

Select this option to enlarge the screen to fit the aspect ratio of the font. This only affects viewing and will not impact coordinate definitions in fields or transitions.

Show only true type fonts

Select this option to display only true type fonts in the font list.

Show only fixed pitch fonts

Select this option to display only fixed-pitch fonts in the font list. Fixed pitch fonts often dislay better than true type fonts.

Undo Stack Depth

The number of actions that Interaction Host Recorder will allow you to undo.

Other pages in the Preferences dialog:

- Preferences (General page)
- Preferences (Host Fields page)
- Preferences (Keyboard Map page)

Preferences (Keyboard Map page)

The Preferences dialog is where you can change how Interaction Host Recorder behaves. On the Keyboard Map page you can re-map your keyboard to emulate a host terminal keyboard. Keep in mind that if you re-map function keys, such as F1, certain Interaction Host Recorder functions may be available only from the menu. These settings only affect how Interaction Host Recorder behaves while you are editing, and they do not change any values within the host profiles.

Key combination to map

This field contains the value of any keys you press on the keyboard. These are the keys that will be mapped.

Host Key

The key that will be written to the host screen when you press the key combination from the previous field.

Example: Mapping the text "exit" to Ctrl-X

1. On the Keyboard Map page, place your cursor in the **Key combination to map** field.
2. Press and hold the Ctrl key, then press the X key. "Control X" appears in this **Key combination to map** field.
3. Place your cursor in the **Host Key** field.
4. Type "exit".
5. Click the **Assign** button. Now, if the cursor is in an unprotected field and you press Ctrl-X, the word exit appears.

Other pages in the Preferences dialog:

- Preferences (General page)
- Preferences (Appearance page)
- Preferences (Host Fields page)

Recording Settings

Command Recording Settings (SendKey page)

Use the Command Recording Settings dialog to configure transition recording behavior. On the SendKey page you can configure how your key presses are converted to transition commands.

Record Command option

Select this option if you want key presses recorded as transition commands. Clear this option if you do not want key presses recorded as transition commands. If this option is selected, the following three options are available.

Record an individual command for each key option

If this option is selected, then each key press is recorded in its own transition command. Having individual transition commands for each keystroke may offer more control, but it can lead to very large transitions.

Combine non-mnemonic keys into one command

If this option is selected, then all non-mnemonic keys are combined into one command until a mnemonic key is pressed. For example, if you pressed A,B,C, Down-Arrow, E, and F, the ABC would all be in one command, down-arrow would be in one command, and EF would be in one command.

Combine all keys into one command

If this option is selected, then all keys that you press are combined into one key command. For example, if you pressed A, B, down-arrow, C, then the command would contain AB down-arrow C.

Other pages in the Preferences dialog:

- Command Recording Settings (SetCursor page)
- Command Recording Settings (WaitForCursor page)
- Command Recording Settings (WaitForRTS page)

Command Recording Settings (SetCursor page)

Use the Command Recording Settings dialog to configure transition recording behavior. On the SetCursor page you can configure how your cursor movements are converted to transition commands.

Record Command option

Select this option if you want cursor movements recorded as transition commands. Clear this option if you do not want cursor movements recorded as transition commands. If this option is selected, the following two options are available.

Discard immediate cursor movements

If this option is selected, then multiple cursor movements using the mouse are interpreted as one cursor movement to the final location of the cursor. If this option is selected then each cursor movement becomes a separate command.

Convert keyboard cursor movements into 'SetCursor'

If this option is selected, then cursor movements you make on the screen become SetCursor commands in the transition command. This has the effect of moving the cursor to an absolute position instead of a relative position.

Other pages in the Preferences dialog:

- Command Recording Setting (SendKey page)
- Command Recording Settings (WaitForCursor page)
- Command Recording Settings (WaitForRTS page)

Command Recording Settings (WaitForCursor page)

Use the Command Recording Settings dialog to configure transition recording behavior. On the WaitForCursor page you can configure how Interaction Host Recorder adjusts wait for cursor commands. See the example at the end of this topic for a graphical representation of the timeout calculation.

Record Command option

Select this option if you want wait for cursor commands automatically added to your transitions while you are recording.

Don't add command for cursor movements to (1,1)

If this option is selected, wait for cursor commands are added when the cursor moves to position (1,1).

Timeout Calculations

Configure these settings to add time to the timeout of a wait for cursor transition command. For example, while recording your transition commands, a wait for cursor command takes one second. By configuring the timeout calculation settings, this time could be automatically changed to 3 seconds.

Multiplier

The value by which you will multiply the wait for cursor time, multiplied by .1. For example, if you want to automatically double the amount of time in the wait for cursor transition command, you would type twenty in this field: 20 x .1 = 2. If the wait for cursor time while you are recording is 1 second, then the time value in the resulting transition command will be 2 seconds. If the wait for cursor time while you are recording is 2 seconds, then the time value in the resulting transition command will be 4 seconds.

Caution: When setting the Multiplier value, take into account that slight variations in the host response time could cause the Wait for Cursor to time out. Setting the multiplier value to 30 is generally sufficient, but thousands of handlers running simultaneously may cause an unforeseen delay in host response. If your handlers are failing, check the handler logs to verify that your multiplier values are sufficient.

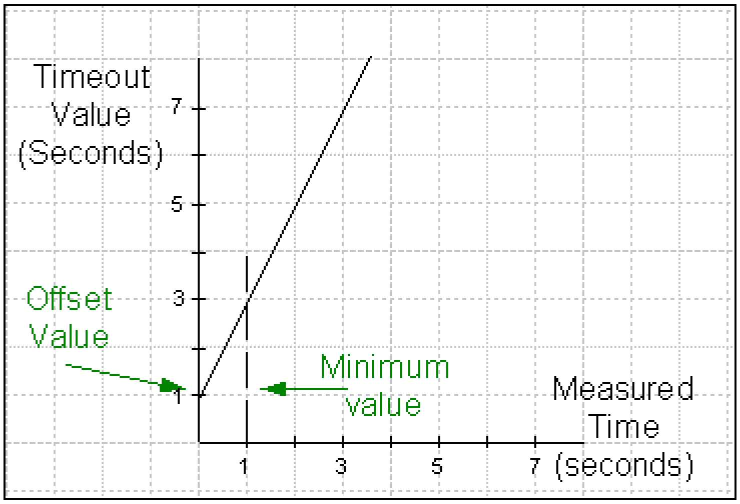
Offset

The number of milliseconds to add to all wait for cursor commands. Type 1000 to add 1 second to all wait for cursor commands.

Minimum value

The minimum value to use in wait for cursor transition commands. If you typed 1000 in this field, all recorded wait for cursor commands will last at least 1 second before timing out.

Example: In the following figure, if the recorded time is 2 seconds, the multiplier is 20, and the offset is 1000 ms, then the wait for cursor transition command will have a time of five seconds.



Other pages in the Preferences dialog:

- Command Recording Setting (SendKey page)
- Command Recording Settings (SetCursor page)
- Command Recording Settings (WaitForRTS page)

Command Recording Settings (WaitForRTS page)

Use the Command Recording Settings dialog to configure transition recording behavior. On the WaitForRTS page you can configure how Interaction Host Recorder adjusts wait for RTS commands. See the example at the end of this topic for a graphical representation of the timeout calculation.

Record Command option

Select this option if you want wait for RTS commands automatically added to your transitions while you are recording.

Timeout Calculations

Configure these settings to add time to the timeout of a wait for RTS transition command. For example, while recording your transition commands, a wait for RTS command takes 2 second. By configuring the timeout calculation settings, this time could be automatically changed to 5 seconds.

Multiplier

The value by which you will multiply the wait for RTS time, multiplied by .1. For example, if you want to automatically double the amount of time in the wait for RTS transition command, you would type twenty in this field: 20 x .1 = 2. If the wait for RTS time while you are recording is 1 second, then the time value in the resulting transition command will be 2 seconds. If the wait for RTS time while you are recording is 2 seconds, then the time value in the resulting transition command will be 4 seconds.

Caution: When setting the Multiplier value, take into account that slight variations in the host response time could cause the Wait for RTS to time out. Setting the multiplier value to 30 is generally sufficient, but thousands of handlers running simultaneously may cause an unforeseen delay in host response. If your handlers are failing, check the handler logs to verify that your multiplier values are sufficient.

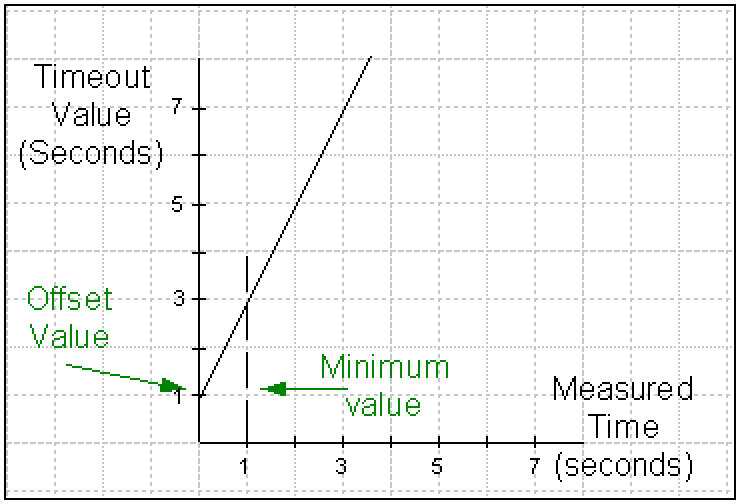
Offset

The number of milliseconds to add to all wait for RTS commands. Type 1000 to add 1 second to all wait for cursor commands.

Minimum value

The minimum value to use in wait for cursor transition commands. If you typed 1000 in this field, all recorded wait for cursor commands will last at least 1 second before timing out.

Example: In the following figure, if the recorded time is 2 seconds, the multiplier is 20, and the offset is 1000 ms, then the wait for RTS transition command will have a time of five seconds.



Other pages in the Preferences dialog:

- Command Recording Setting (SendKey page)
- Command Recording Settings (SetCursor page)
- Command Recording Settings (WaitForCursor page)

Publishing Host Profiles

Publishing Host Profiles

Publishing is the process by which the host profile is loaded onto the Host Server . Once a host profile is published, Host tools in handlers can instruct the Host Server to perform actions based on the information in the profile.

Caution: When you publish a host profile, the data within that profile is loaded on to the Host Server. When opening a host profile for editing, you have the choice of opening the host profile from the Host Server, or from the machine where you saved a copy. If you choose to open and edit the profile loaded on the Host Server, keep in mind that it is possible that changes made by others could cause the host profile on the server to be different than the host profile saved locally. In many cases, you may want to open your local copy and republish any changes you make. This will ensure that both versions are identical.

To publish a host profile:

1. Open the Profile you want to publish.
2. From the File menu, choose **Publish**.
   The file is published to the Host Server.

Manage Host Profiles dialog

The Manage Host Profiles dialog displays all the host profiles currently published on your Host Server. From this dialog you can perform the following functions:

Add

Allows you to select and publish a host profile, this adding it to the list of published host profiles.

Delete

Removes the selected host profile from the server and the Published Profiles list.

Open

Opens the selected host profile in Interaction Host Recorder.

Glossary

Handler (definition)

A program built in the Interaction Designer program that performs some action or actions in response to the occurrence of some event. A handler is a collection of steps (called tools) organized and linked to form a logical flow of actions and decisions. Handlers are similar in structure to a detailed flowchart. Using Host tools in handlers allows a handler to send commands to the Host Server.

Host (definition)

A computer to which one connects with a terminal emulator. A terminal emulator is a program that allows a computer to act as a particular brand of terminal when accessing a host. Interaction Host Recorder has a built-in terminal emulator that you use to create host profiles. Interaction Host Recorder can connect to two types of hosts: Mainframes and AS/400s.

Host Profile (definition)

Host profiles are the instructions that the Host Server uses to open and navigate through a host session. The Interaction Host Recorder is the tool you use to create these instructions.

Host Server (definition)

The Host Server is the IC server subsystem that executes the host profiles. Handlers can send commands to the Host Server; the Host Server reads the host profile, navigates through screens, and reads and writes information.

Host Tools (definition)

Tools are like function calls within a handler. The Host tools enable handlers to send commands to the Host Server , which in turn reads the host profile and performs the requested action on the host. For more information on Host tools and handlers, see the Interaction Designer online help.

See Host Tool Overview for more information on how the host tools interact with the host profiles.

Mnemonic keys (definition)

Keyboard mnemonics are ASCII characters that represent non-data keystrokes such as special function keys. For example, by default the down-arrow key on your keyboard sends a @V command to the host screen. In a transition command, if only one mnemonic key is sent to the host screen, Interaction Host Recorder will attempt to identify the label associated with that mnemonic. For example, if you press the down arrow key once in a command, Interaction Host Recorder displays the text "Cursor Down" in the transition command. If you press the down arrow key twice in one command, @V@V appears. This is due to a limitation in Interaction Host Recorder's ability to parse complex mnemonic sequences.

The following list shows all the mnemonic keys mapped my default, You can change the mnemonic values and their labels on the Keyboard Map page in the Preferences dialog.

Mnemonic Mappings

Meaning	Mnemonic	Mainframe	AS/400
@	@@	X	X
Alt	@A	X	
Alt Cursor	@e@1	X	X
Attention	@A@Q	X	X
Backspace	@<	X	X
Backtab (Left Tab)	@B	X	X
Clear	@C	X	X
Cmd (function) Key	@A@Y		X
Cursor Down	@V	X	X
Cursor Left	@L	X	X
Cursor Right	@Z	X	X
Cursor Select	@e@1	X	
Cursor Up	@U	X	X
Delete	@D	X	X
Dup	@S@x	X	X
End	@q		X
Enter	@E	X	X
Erase EOF	@F	X	X
Erase EOL	@e@d		X
Erase Input	@A@F	X	X
Field Exit	@A@E		X
Field Mark	@S@y	X	
Field -	@A@-		X
Field +	@A@+		X
Help	@H		X
Hexadecimal	@A@X		X
Home	@0 (zero)	X	X
Insert	@I	X	X
Insert Toggle	@A@I		X
Local Help	@e@l		X
Local Print	@P		X
Local Help Cursor	@e@m	X	X
Left Tab (Back Tab)	@B	X	X
New Line	@e@n	X	X
Page Up	@u		X

Meaning	Mnemonic	Mainframe	AS/400
Page Down	@v		X
Print (PC)	@A@t	X	X
Print Screen	@A@T	X	
Record Backspace	@A@<		X
Reset	@R	X	X
Right Tab (Tab)	@T	X	X
Shift	@S	X	
Sys Request	@A@H	X	X
Tab (Right Tab)	@T	X	X
Test	@A@C		X
PA1	@x	X	
PA2	@y	X	
PA3	@z	X	
PA4	@+	X	
PA5	@%	X	
PA6	@&	X	
PA7	@'	X	
PA8	@(	X	
PA9	@)	X	
PA10	@*	X	
PF1/F1	@1	X	X
PF2/F2	@2	X	X
PF3/F3	@3	X	X
PF4/F4	@4	X	X
PF5/F5	@5	X	X
PF6/F6	@6	X	X
PF7/F7	@7	X	X
PF8/F8	@8	X	X
PF9/F9	@9	X	X
PF10/F10	@a	X	X
PF11/F11	@b	X	X
PF12/F12	@c	X	X
PF13/F13	@d	X	X
PF14/F14	@e	X	X
PF15/F15	@f	X	X
PF16/F16	@g	X	X

Meaning	Mnemonic	Mainframe	AS/400
PF17/F17	@h	X	X
PF18/F18	@i	X	X
PF19/F19	@j	X	X
PF20/F20	@k	X	X
PF21/F21	@l	X	X
PF22/F22	@m	X	X
PF23/F23	@n	X	X
PF24/F24	@o	X	X

Publishing (definition)

The process by which a host profile is made available on the Host Server . Once a host profile is published, handlers can use the host profile to send commands to the host.

Screen (definition)

A host screen is the current contents of a connected terminal emulation window. You should assign names and validation rules to all screens that you will use in transition.

Screen Fields (definition)

Screen fields are the fields that Interaction Host Recorder can read when a screen is opened. They are defined by the person who programmed the screens, and Interaction Host Recorder lists them in the Host Screen Fields window.

Screen fields can be protected or unprotected:

- **Protected screen fields**
  These are read-only fields in which data is displayed. If you create a user defined field from a protected field, a handler can only read the data. You cannot change the value in a protected screen field. Think of protected fields as fields that can be read but not changed.

- **Unprotected screen fields**
  These are read/write fields into which data can be written. If you create a user defined field from an unprotected screen field, a handler can write data to the screen. An example of an unprotected field is a Choice where you type a menu entry to change screens.

See Host Screen Fields window for more information on screen fields.

Screen Type (definition)

Screen type is simply the number of unprotected fields in a screen. If a screen only has one unprotected field, then the screen type will be 1.

Screen ID (definition)

Each screen has a unique identifier generated mathematically from the number, position, and length of the fields. If you know that the fields will always be in the same place and of the same length, you can rely on this value being the same each time the screen is loaded. You can use screen ID even if some of the values within fields are going to change. For example, if there is a time stamp located in the top left corner, and that field is always the same length, the screen will keep it's unique ID even if the time of day is different.

Transition (definition)

Transitions are the mechanism by which Host Server navigates from screen to screen. You can create transitions in the Transition Commands window. Transitions contain transition commands.

There are two types of transitions: **normal** and **start up**

- **Normal transitions** navigate from screen A to screen B. In the Transition list, normal transitions look like this:

  ScreenA ------------> ScreenB

- **Start up transitions** are started just after the host connection is successfully established. It contains instructions for opening the first screen. A profile may contain only one start up transition. In the transition list, the start up transition looks like this:

  {connected} ----------> ScreenA

Note: When creating transitions, remember that they do not have to be to the next screen. A transition can navigate though several screens before it arrives at the target screen. Be sure to add validation transition commands if you are building transition that cross multiple screens.

See Transition Commands window for more information on transitions and transition commands.

Transition Commands (definition)

Transition commands are the instructions that make up a transition. You can record a series of transition commands automatically by clicking the Start Recording button, or you can insert transition commands manually in the Transitions Commands window.

See Transition Commands window for more information on creating transition commands.

User Defined Fields (definition)

User defined fields are fields you create to allow handlers to write information to and from screens. User defined fields enable handlers to navigate and process information within screens just like a person would. You must create user defined fields, as opposed to screen fields that Interaction Host Recorder creates automatically when a screen is opened.

See Screen Definition window for more information on creating user defined fields.

Validation Rules (definition)

Validation rules contain criteria for identifying the currently loaded screen. When a screen appears, Interaction Host Recorder checks the validation rules to determine which screen is loaded. Validation rules can include screen ID, the position of text on the screen, and several other defining characteristics of a screen.

See Validation Rules for more information on the types of validation rules and how you can configure them.