



QAD Enterprise Applications

Installation Guide
QAD Integrated Customization
Toolkit

70-3260-4.1.14-Rev1
QAD ICT 4.1.14
June 2013

This document contains proprietary information that is protected by copyright and other intellectual property laws. No part of this document may be reproduced, translated, or modified without the prior written consent of QAD Inc. The information contained in this document is subject to change without notice.

QAD Inc. provides this material as is and makes no warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. QAD Inc. shall not be liable for errors contained herein or for incidental or consequential damages (including lost profits) in connection with the furnishing, performance, or use of this material whether based on warranty, contract, or other legal theory.

QAD and MFG/PRO are registered trademarks of QAD Inc. The QAD logo is a trademark of QAD Inc.

Designations used by other companies to distinguish their products are often claimed as trademarks. In this document, the product names appear in initial capital or all capital letters. Contact the appropriate companies for more information regarding trademarks and registration.

Copyright ©2013 by QAD Inc.

IntegratedCustomizationToolkit_UI_v04114-Rev1.pdf/mat/mat

QAD Inc.

100 Innovation Place
Santa Barbara, California 93108
Phone (805) 566-6000
<http://www.qad.com>

Contents

Change Summary	v
Chapter 1 Overview	1
Introduction	2
QAD ICT	2
Chapter 2 Preparation for the Installation	3
Initial Considerations	4
Upgrades from Earlier Versions	4
Chapter 3 Installation	5
Standard Edition	6
Extract Software Package	6
Update the Standard Edition Startup Programs	6
Update Additional Startup Programs	7
Enterprise Edition Earlier than 2011 EE	9
Extract Software Package	9
Update the Enterprise Edition Startup Programs (before 2011 EE)	9
Update Additional Startup Programs	10
Enterprise Edition 2011 EE, 2011.1 EE, and 2012 EE	11
Extract Software Package	11
Update the Enterprise Edition Startup Programs (2011 EE, 2011.1 EE, and 2012 EE)	11
Update Additional Startup Programs	12
Enterprise Edition 2012.1 EE and Later	13
Chapter 4 Compiling and Completing the Installation	15
Compilation Considerations	15
Non-domained Versions	15
Setup File	16
Standard Edition	16
Enterprise Edition Before 2011 EE	18
Enterprise Edition 2011 EE, 2011.1 EE, and 2012 EE	19
UNIX	19

Windows	20
Propath Settings	20
Enterprise Edition 2012.1 EE and Later	21
UNIX	21
Windows	22
Propath Settings	22
Additional Tasks for All Versions	23
Update Runtime Scripts and Properties Files	23
Start QAD Enterprise Applications	23
Define QAD ICT Licensing	26
Set Up QAD ICT Control Table	26
Validate that QAD ICT Is Working	27

Change Summary

The following table summarizes significant differences between this document and previous versions.

Date/Version	Description	Reference
June 2013/QAD ICT 4.1.14 Rev 1	Added separate compile instructions for Windows on Enterprise Edition	page 20 page 22
April 2013/QAD ICT 4.1.14	Numerous changes throughout	--
November 2012/QAD ICT 4.1	Provided additional information on updating startup programs for 2011-2012 EE	page 11
	Provided additional information on 2012.1 EE install	page 13
October 2012/QAD ICT 4.1	Updated to reference version 4.1	throughout
	Removed information on transferring software package	throughout
	Added information on using QAD ICT with later versions of QAD Enterprise Applications	throughout
July 2012/QAD ICT 4.0	Corrected message range reference to say 39100-39199	page 4
	Removed reference to icacc.p	page 26
	Replaced screen capture of ICT Control Table	page 26
June 2012/QAD ICT 4.0	Removed "Upgrade" chapter; replaced with upgrade instructions as part of normal installation	various
May 2012/QAD ICT 4.0	First formal release of document	

Overview

This section contains basic topics that you should understand before attempting a QAD ICT installation or upgrade.

Introduction 2

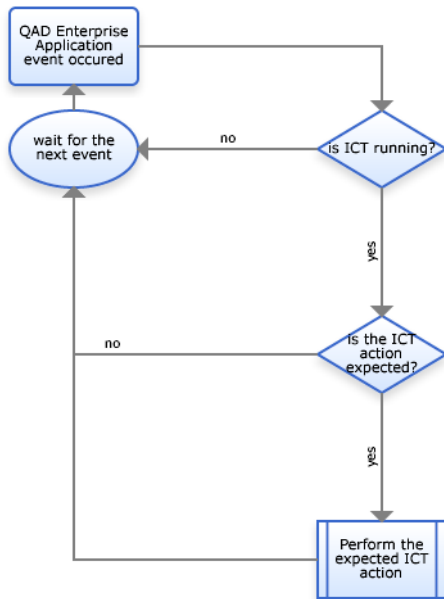
QAD ICT 2

Introduction

This chapter provides an overview of QAD ICT and describes how QAD ICT complements QAD Enterprise Applications.

QAD ICT

The QAD Integrated Customization Toolkit enables you to design and develop customizations in a non-intrusive way, which means eliminating or limiting the changes in the original standard applications. The following figure is an overview of ICT processing. See *User Guide: QAD Integrated Customization Toolkit* for more information.



Preparation for the Installation

This chapter includes information you can use when preparing to install QAD ICT.

Initial Considerations 4

Upgrades from Earlier Versions 4

Initial Considerations

QAD ICT is a service product and should be installed as any other customization you receive from QAD Services. In this installation guide, it is installed in the \$QADICT layer.

QAD recommends that QAD ICT should be first installed in the test environment and (if approved) then in the production environment.

Note QAD ICT uses message numbers from 39100 to 39199. It is highly recommended that you delete all custom messages in the 39100-39199 range because they will be overwritten during the ICT installation process.

Upgrades from Earlier Versions

QAD ICT 4.1 is backward compatible with earlier versions, but it is recommended that after the installation of QAD ICT 4.1, the developers use the new features and functions in the latest version.

Note If you use the task utility in your current version of ICT, you should close all WIP tasks before the upgrade process. Otherwise, they will automatically be closed by ICT.

Note After upgrade you must configure ICT Control Table (90.24.24) because the whole setup will be overwritten

The following list summarizes the upgrade from each earlier version. Perform the installation as described in this guide. All additional changes required to upgrade from an earlier version of QAD ICT are marked as **UPGRADE NOTE**.

- **QAD ICT 1.x.** The only required step, after installing QAD ICT 4.1 in a QAD ICT 1.0 environment, is to execute a utility program that deletes the old menu structure of QAD ICT 1.0.
- **QAD ICT 2.x.** The only required step, after installing QAD ICT 4.1 in a QAD ICT 2.0 (SP1) environment, is to execute a utility program that reorders some QAD ICT data.
- **QAD ICT 3.x.** The only required step, after installing QAD ICT 4.1 in a QAD ICT 3.0 environment, is to execute (one time) the utility programs that complete new data required by the upgraded functionality of ICT tasks and Frame & Field properties.
- **QAD ICT 4.0.** No additional steps are required.

Installation

This chapter describes QAD ICT installation tasks, including the changes to be made in the startup programs of the Standard and Enterprise Edition releases of QAD Enterprise Applications. It is assumed that QAD Development Services has already done this or the changes will be made locally.

Important If this has not been supplied to you (in addition to the QAD ICT software), please consult QAD Services. QAD ICT cannot be activated without these changes.

Standard Edition 6

Enterprise Edition Earlier than 2011 EE 9

Enterprise Edition 2011 EE, 2011.1 EE, and 2012 EE 11

Enterprise Edition 2012.1 EE and Later 13

Standard Edition

This section describes installation tasks for Standard Edition.

Extract Software Package

Extract the software from the `<release_number>_1` directory in the `$QADICTINSTALL` directory and make sure that all resources are stored as follows:

- xrc contents in `$QADICT/xrc`
- src contents in `$QADICT/src` (optional)
- utcompil.wrk in `$QADICT`
- Data contents in `$QADICT/data`

Copy the following menu program file:

```
cp $QADICTINSTALL/SE_Files/icmenu.p.se $QADICT/xrc/icmenu.p
```

Update the Standard Edition Startup Programs

This paragraph describes the changes to be made in the startup programs of the Standard Edition releases of QAD Enterprise Applications. Please modify the follow programs as instructed.

Note If these programs have been modified and are stored in directories before the `$QADICT` layer in the Propath (for example, PATCH, CUST), the described changes should be made in the modified programs, not in standard ones.

Make the following change in the custom or standard version of `mfl.p` of the QAD Enterprise Applications release the customer is using. Add the QAD ICT include just before the call to `mfla.p`.

```
/* QAD ICT - Integrated Customization Toolkit */
{icmf.i}

/* NOW CALL THE TRANSLATABLE PORTION OF THE (OLD) MF1.P */
{gprunl.i "mfla.p"}

PROCEDURE registerReasonMessage:
```

Make the following change in the custom or standard version of `mfwb01a.p` of the QAD Enterprise Applications release the customer is using. Please add the QAD ICT include just before run `set-session-formats`.

```
/* QAD ICT - Integrated Customization Toolkit */
{icmf.i}

run set-session-formats in sess-context-hdl
(input global_userid).

/*Check license usage*/
{gprunp.i
"lvgenpl" "p"
"validateUserCountExceeded_HTML"
"(input 'MFG/PRO', output licenseMsgNum, output licenseMsgDesc)" }

```

For QAD SE 2010+ .NET UI 2.9.1+ versions:

```
/* QAD ICT - Integrated Customization Toolkit */
{icmf.i}
run set-session-formats in mfwb02handle()
(input global_userid).
```

```

/*Check license usage*/
{gprunp.i
"lvgenpl" "p"
"validateUserCountExceeded_HTML"
"(input 'MFG/PRO', output licenseMsgNum, output licenseMsgDesc)" }

```

Update Additional Startup Programs

As needed, modify the startup programs for additional QAD Enterprise Applications as shown.

QXtend

Make the following change in the custom or standard version of `mfwb01aa.p` of the QAD Enterprise Applications release the customer is using. Please add the QAD ICT include just before the do transaction for `lvucap.p`.

```

/* QAD ICT - Integrated Customization Toolkit */
{icmf.i}

/* Log the transaction for the pricing */
do transaction:
  {gprunp.i "lvucap.p"
  "(input ''qxtend'', input '')"}
end.

```

QGen

Make the following change in the custom or standard version of `runProgramMapper.p` of the QXtend release the customer is using. The change consists of two changes; just before the on leave trigger definition and at the end of the leave trigger definition.

The optional language parameter of `icqgen.i` should be the default language that is being used for that customer (in most cases, that will be US).

```

/* Reset everything ready for a new run through. */
run newRunThrough in pHandle.

/* QAD ICT - Integrated Customization Toolkit */
{icqgen.i &LANGUAGE="us"}

/***** TRIGGERS *****/
/* On leaving a field update the data for the field */
on leave anywhere do:
  /*
  * Work out a valid focus handle. This will usually just be the
  ...
          else
            message "Invalid handle pHandle".
          end.
        end.
      end.
    end.

/* QAD ICT - Integrated Customization Toolkit */
{icttrglv.i}
end.

```

Note Make sure that the compiled object for `runProgramMapper` is saved in the proper directory and that the QGen script will first pick up this customized version.

AIM API

Make the following changes in the include `whgptrrf.i` of the AIM release the customer is using. Add the additional code as shown below.

```

/*V8:ConvertMode=NoConvert*/

/* QAD ICT - Integrated Customization Toolkit */
{icdef.i &AIMTRIGGERS=true}

/*****
/* GLOBAL GO TRIGGER */
/*****
on go anywhere do:
    {icttrggo.i} /*QAD ICT*/
    run proc_ui_check no-error.
end.

/*****
/* GLOBAL LEAVE TRIGGER */
/*****
on leave anywhere do:
    {icttrglv.i} /*QAD ICT*/
    run proc_ui_check no-error.
end.

on entry anywhere do:
    {icttrgen.i} /*QAD ICT*/
end.

/*****
/* ASSIGN SR_QTY */

```

Note Make sure that you recompile `whaimapi.p` and `whapist2.p`, as these programs use the updated `whgptrrf.i` and are called in the AIM startup programs for Character, GUI, Desktop, and .NET UI.

RF Terminals

Make the following change in the custom or standard version of `whmf1.p` of the QAD Enterprise Applications release the customer is using. Add the QAD ICT include just before the call to `whmf1a.p`.

```

/* QAD ICT - Integrated Customization Toolkit */
{icmf.i}

{gprun1.i "whmf1a.p"}

{whgpevip.i} /*ENVIRONMENT PROCEDURES*/

```

Add the following lines at the end of the ICT `utcompil.wrk` list:

```

whmf1.p
whaimapi.p
whapist2.p
runProgramMapper.p
mfwb01aa.p
mfwb01a.p
mf1.p

```

Enterprise Edition Earlier than 2011 EE

This section describes installation tasks for Enterprise Edition versions before 2011 EE.

Extract Software Package

Extract the software from the `<release_number>_1` directory in the `$QADICTINSTALL` directory and make sure that all resources are stored as follows:

- xrc contents in the `$QADICT/xrc`
- src contents in the `$QADICT/src` (optional)
- utcompil.wrk in the `$QADICT`
- Data contents in the `$QADICT/data`

Update the Enterprise Edition Startup Programs (before 2011 EE)

This section describes the changes to be made in the startup programs of the Enterprise Edition releases of QAD Enterprise Applications. Please modify the follow programs as instructed.

Note If these programs have been modified and are stored in directories before the `$QADICT` layer in the Propath (for example, PATCH, CUST), the described changes should be made in the modified programs, not in standard ones.

Make the following change in the custom or standard version of `whapish.p` of the Enterprise Edition release the customer is using. Add the QAD ICT include just before the variable definitions.

```
{mfdeclre.i}

/* QAD ICT - Integrated Customization Toolkit */
{icdef.i &AIMTRIGGERS=false}

define variable LVApiGblHandle as handle no-undo.
define variable LVApiTrfHandle as handle no-undo.
```

Also in the custom or standard version of `whapist.p` of the Enterprise Edition release the customer is using, add QAD ICT at the end of the program.

```
/* Shut down AIM triggers. */
{gprun.i "whapish.p"}
end.

/* QAD ICT - Integrated Customization Toolkit */
{icmf.i}
```

Make the following changes in the include `whgptrrf.i` of the Enterprise Edition release the customer is using. Add the additional code as shown below.

```
/*V8:ConvertMode=NoConvert*/

/* QAD ICT - Integrated Customization Toolkit */
{icdef.i &AIMTRIGGERS=true}

/*****/
/* GLOBAL GO TRIGGER */
/*****/
on go anywhere do:
  {icttrggo.i} /*QAD ICT*/
  run proc_ui_check no-error.
end.

/*****/
```

10 Installation Guide — Integrated Customization Toolkit

```
/* GLOBAL LEAVE TRIGGER */
/*****/
on leave anywhere do:
    {icttrglv.i} /*QAD ICT*/
    run proc_ui_check no-error.
end.

on entry anywhere do:
    {icttrgen.i} /*QAD ICT*/
end.

/*****/
/* ASSIGN SR_QTY */
```

Note Make sure that you recompile whgblmgr.p, as this program uses the updated whgprrf.i.

Update Additional Startup Programs

As needed, modify the startup programs for additional QAD Enterprise Applications as shown.

QXtend

Make the following change in the custom or standard version of mfwb01aa.p of the QAD Enterprise Applications release the customer is using. Add the QAD ICT include just before the do transaction for lvucap.p.

```
/* QAD ICT - Integrated Customization Toolkit */
{icmf.i}

/* Log the transaction for the pricing */
do transaction:
    {gprun.i "lvucap.p"
        "(input 'qxextend', input '')"}
end.
```

QGen

Make the following change in the custom or standard version of runProgramMapper.p of the QXtend release the customer is using.

Note When using QGen, the Ctrl+O key will show the QGen menu and not the QAD ICT developers menu, if the QAD ICT menu accelerator is set to Ctrl+O.

The change consists of two changes; just before the on leave trigger definition and at the end of the leave trigger definition.

The optional language parameter of icqgen.i should be the default language that is being used for that customer (in most cases, that will be US).

```
/* Reset everything ready for a new run through. */
run newRunThrough in pHandle.

/* QAD ICT - Integrated Customization Toolkit */
{icqgen.i &LANGUAGE="us"}

/***** TRIGGERS *****/
/* On leaving a field update the data for the field */
on leave anywhere do:
    /*
    * Work out a valid focus handle. This will usually just be the
    ...
        else
            message "Invalid handle pHandle".
```

```

        end.
    end.
end.
/* QAD ICT - Integrated Customization Toolkit */
{icttrglv.i}
end.

```

Note Make sure that the object for `runProgramMapper` is saved in the base directory and that the QGen script will first pick up this customized version.

Add the following lines at the end of the ICT `utcompil.wrk` list:

```

whgblmgr.p
whapish.p
whapist.p
runProgramMapper.p

```

Enterprise Edition 2011 EE, 2011.1 EE, and 2012 EE

This section describes the installation tasks for those versions of Enterprise Edition.

Extract Software Package

Extract the software from the `<release_number>_2` directory in the `$QADICTINSTALL` directory and make sure that all resources are stored as follows:

- xrc contents in the `$QADICT/xrc`
- src contents in the `$QADICT/src` (optional)
- `utcompil.wrk` in the `$QADICT`

Update the Enterprise Edition Startup Programs (2011 EE, 2011.1 EE, and 2012 EE)

This section describes the changes to be made in the startup programs of those Enterprise Edition releases of QAD Enterprise Applications. Please modify the following programs as instructed.

Note If these programs have been modified and are stored in directories before the `$QADICT` layer in the `Propath` (for example, `PATCH`, `CUST`), the described changes should be made in the modified programs, not in standard ones.

Make the following change in the custom or standard version of `whapish.p` of the Enterprise Edition release the customer is using. Add the QAD ICT include just before the variable definitions.

```

{mfsubdirs.i}
{us/bbi/mfdeclre.i}

/* QAD ICT - Integrated Customization Toolkit */
{us/bbi/icdef.i &AIMTRIGGERS=false}

define variable LVApiGblHandle as handle no-undo.
define variable LVApiTrfHandle as handle no-undo.

```

Also in the custom or standard version of `whapist.p` of the Enterprise Edition release the customer is using, add the QAD ICT at the end of the program.

```

/* Shut down AIM triggers. */
{us/bbi/gprun.i "whapish.p"}
end.

```

```
/* QAD ICT - Integrated Customization Toolkit */
{us/bbi/icmf.i}
```

And at the beginning of the program:

```
{mfsubdirs.i}
{us/bbi/mfdeclre.i}

define variable LVApiGblHandle as handle no-undo.
```

Make the following changes in the include `whgptrrf.i` of the Enterprise Edition release the customer is using (if available). Add the additional code as shown below.

```
/* QAD ICT - Integrated Customization Toolkit */

{mfsubdirs.i}
{us/bbi/icdef.i &AIMTRIGGERS=true}

/*****/
/* GLOBAL GO TRIGGER */
/*****/
on go anywhere do:
    {us/bbi/ictrggo.i} /*QAD ICT*/
    run proc_ui_check no-error.
end.

/*****/
/* GLOBAL LEAVE TRIGGER */
/*****/
on leave anywhere do:
    {us/bbi/ictrglv.i} /*QAD ICT*/
    run proc_ui_check no-error.
end.

on entry anywhere do:
    {us/bbi/ictrgen.i} /*QAD ICT*/
end.

/*****/
/* ASSIGN SR_QTY */
```

Note Make sure that you recompile `whgblmgr.p`, as this program uses the updated `whgptrrf.i`. The include file `whgptrrf.i` must be stored in the `xrc/us/wh/` directory.

Update Additional Startup Programs

As needed, modify the startup programs for additional QAD Enterprise Applications as shown.

QGen

Make the following change in the custom or standard version of `runProgramMapper.p` of the QXtend release the customer is using.

Note When using QGen, the Ctrl+O key will show the QGen menu and not the QAD ICT developers menu, if the QAD ICT menu accelerator is set to Ctrl+O.

Copy `runProgramMapper.p` into `$QADICT`. Compile the file into the same directory.

The change occurs in two places; just before the on leave trigger definition and at the end of the leave trigger definition.

The optional language parameter of `icqgen.i` should be the default language which is being used for that customer (in most cases, that will be US).

```
/* Reset everything ready for a new run through. */
run newRunThrough in pHandle.
```

```

/* QAD ICT - Integrated Customization Toolkit */
{us/bbi/icqgen.i &LANGUAGE="us"}

/***** TRIGGERS *****/
/* On leaving a field update the data for the field */
on leave anywhere do:
  /*
   * Work out a valid focus handle. This will usually just be the
   ...
           else
               message "Invalid handle pHandle".
           end.
       end.
   end.

   /* QAD ICT - Integrated Customization Toolkit */
   {us/bbi/icttrglv.i}
end.

```

Note Make sure that the object for `runProgramMapper` is saved in the base directory and that the QGen script will first pick up this customized version.

Copy Changed Programs to a Proper Directory

Create the `$QADICT xrc/us/wh` subdirectory for modified EE startup programs:

```

mkdir $QADICT/xrc/us (if necessary)
mkdir $QADICT/xrc/us/wh

```

Make sure all modified `wh*` programs are in `$QADICT/xrc/us/wh`.

Add the following lines at the end of the ICT `utcompil.wrk` list:

```

runProgramMapper.p
us/wh/whapish.p
us/wh/whapist.p
us/wh/whgblmgr.p

```

Note `runProgramMapper.p` must be compiled into `$QADICT`. Directory `$QADICT/tools` must not exist.

Note If `us/wh/whgblmgr.p` is not available, do not add it to the ICT `utcompil.wrk` list.

Enterprise Edition 2012.1 EE and Later

Extract the software from the `<release_number>_2` directory in the `$QADICTINSTALL` directory and make sure that all resources are stored as follows:

- `xrc` contents in `$QADICT/xrc`
- `src` contents in `$QADICT/src` (optional)
- `utcompil.wrk` in `$QADICT`
- Data contents in `$QADICT/data`

Add the following lines at the end of the ICT `utcompil.wrk` list:

```

runProgramMapper.p
us/wh/whapish.p
us/wh/whapist.p
us/wh/whgblmgr.p

```

Note `runProgramMapper.p` must be compiled into `$QADICT`. Directory `$QADICT/tools` must not exist.

Note If `us/wh/whgblmgr.p` is not available, do not add it to the `ICT utcompil.wrk` list.

Compiling and Completing the Installation

This chapter describes the compilation process, which is a part of the correct installation of QAD ICT software, as well as the tasks required to complete the QAD ICT installation.

Note All changes required due to upgrade from an earlier version of QAD ICT are marked as **UPGRADE NOTE**.

Compilation Considerations 15

Non-domained Versions 15

Standard Edition 16

Enterprise Edition Before 2011 EE 18

Enterprise Edition 2011 EE, 2011.1 EE, and 2012 EE 19

Enterprise Edition 2012.1 EE and Later 21

Additional Tasks for All Versions 23

Compilation Considerations

Based on the supplied `utcompil.wrk`, the software must be compiled using an appropriate compiler such as QDT/QDT BatchCompiler or MFGUTIL. If you have any compilation errors, please review and resolve them.

Add the startup programs just modified for ICT to the `$QADICT/utcompil.wrk` list. If they are considered to be compiled in the `$QADICT` layer (neither custom nor patch layer) add them to the `$QADICT/utcompil.wrk` list.

Non-domained Versions

If you install QAD ICT on a non-domained version of QAD Enterprise Applications, read this section first, then perform the compilation described in the appropriate paragraph for the customer's version of QAD Enterprise Applications.

QAD ICT has been developed for eB2.1+ (domain version), but can also be installed on non-domain releases. If this is the case, set QAD ICT to non-domain, review the browse definition, and regenerate these.

Note It is assumed that QAD Services will supply the correct `icdomain.i` and dump file before installing.

Setup File

As QAD ICT by default is set up for domained versions of QAD Enterprise Applications, you must change a setup file before you compile the software.

Copy the `icdomain.i.non-domained.mfg` file from the `/mfg` directory and rename it to `icdomain.i`. Move the file to the xrc layer (thus overwriting the default).

Standard Edition

Launch the main `mfgutil` utility script/shortcut from the `$QADEA` directory. Alternatively, to avoid modifications to the main `MFG/UTIL`, copy `mfgutil` and `mfgutil.ini` to the `$QADICT` folder and adjust the `mfgutil` script/shortcut to run it in the `$QADICT` working directory, then launch this utility from `$QADICT`.

Copy the existing compilation database set used to compile European Accounting or base programs or any other layer, that connects to all required databases (for example, `CompileUS-EA` or `CompileUS`, or `Compile`) and create the `CompileUS-ICT` database set.

Select a Database Set

Set Name	Set Description	Active	Start Parameters
CompileUS-EA	CompileUS-EA Databas	NO	-rereadnolock -c 30 -d mdy -yy
>CompileUS-ICT	CompileUS-EA Databas	NO	-rereadnolock -c 30 -d mdy -yy

<Edit Set > < New Set > <Copy Set > < Help > < Cancel > < OK >

Selected Set Overview

Physical	Description	Host	Service	Path
>mfgempty	Empty Q&D ERP Databa			/ul/qad/ea2010se/db
hlpempty	Empty Q&D ERP Help D			/ul/qad/ea2010se/db
admempty	Empty Q&D ERP Admin			/ul/qad/ea2010se/db
audempty	Empty Q&D ERP Audit			/ul/qad/ea2010se/db

< Edit DB > < New DB > <Delete DB>

Add corresponding database sets for each required language (for example, `CompilePL-ICT`) based on existing databases sets connecting to the language-specific compilation databases.

Compile ICT for US language using the just-created database set and the `$QADICT/utcompil.wrk` compilation list. Make sure that the `Propath` is set properly to point to the `$QADICT` folder on the first position: `ICT,(QXTEND),(MSW/PSW),(WMS),CORE` (all core layers, including `BBI`, `FIN`, `QRA`, `PROXY`).

Compiler Options		R-code Destination	
<input checked="" type="checkbox"/> Verbose <input type="checkbox"/> Silent	<input type="checkbox"/> Generate Compile List File <input checked="" type="checkbox"/> Use Existing Compile List	<input checked="" type="checkbox"/> Staggered QAD ERP Default <input type="checkbox"/> Flat Destination <input type="checkbox"/> Destination Same as Source	
Compile List File: /ul/qad/ea2010se/ict/utcompil.wrk		<Browse>	
Compile Propath: /ul/qad/ea2010se/ict/xrc,/ul/qad/ea2010se/qxtend/xrc,/ul/qad/ea2010se/euroacc/xrc,/ul/qad/ea2010		< Edit >	
Language Code: us [V]		Database Set: CompileUS-ICT [V]	
Destination Directory: /ul/qad/ea2010se/ict		<Browse>	
<Compile >		< Help >	
		< Close >	

```

/ul/qad/ea2010se/ict/xrc
> /ul/qad/ea2010se/ict/xrc
/ul/qad/ea2010se/qxtend/xrc
/ul/qad/ea2010se/euroacc/xrc
/ul/qad/ea2010se/xrc
    
```

```

03/31/12 @ 18:47:22 - Compiling /ul/qad/ea2010se/ict/xrc/mfwb01a.p
03/31/12 @ 18:47:23 - Compiling /ul/qad/ea2010se/ict/xrc/mfwb01aa.p
03/31/12 @ 18:47:23 - Compile Contained 0 Errors.
03/31/12 @ 18:47:23 - Compile Contained 0 Warnings.
03/31/12 @ 18:47:23 - Resetting PROPATH and Disconnecting from Databases.
03/31/12 @ 18:47:23 - End Compile.
Press CLOSE to continue.
    
```

< Close >

```

                                Program Compile Status
Files Processed: 112 of 112                Now Compiling: mfwb01aa.p
Errors Found: 0                            Percentage Complete: 100%
    
```

Note R-code of AIM programs (wh* programs) should not be stored in the standard language directory (/language/wh). It should be compiled directly to the \$QADICT directory.

Do this by adding the following QAD Warehousing (AIM) related programs to the \$QADEA/toplevel.lst file:

- cpdchk.r
- cpdmf.r
- cpdmf1.r
- cpdst2.r

```

cpdstart.r
cpdstop.r
cpdtrig.r
whaimapi.r
whapist2.r
whmf1.r
whtrtxfr.r

```

This will ensure that these QAD Warehousing top-level programs will be compiled to the proper directory during all subsequent compilations done with MFG/UTIL.

Enterprise Edition Before 2011 EE

Compile ICT and EE modified programs using the QDT batch compiler.

Create the `$QADICT/compileICT.ksh` script.

Make sure the `Propath` and other parameters are properly set for an EE environment. Detailed information about how to run the QDT Batch Compiler is available in the existing batch compiler script—for example, `$QADQDT/envs/$QADENV/scripts/01batchCompile.ksh`—or in the `01batchCompile.log` files from the `$QADQDT/logs` directory.

The example below assumes you are installing ICT into `/dr01/qadapps/ict` to an existing 2010 EE installation found in `/dr01/qadapps/qea` with QDT in `/dr01/qadapps/qdt`.

```

#!/bin/ksh
DLC=/dr01/progress/dlc; export DLC
PATH=$PATH:$DLC/bin; export PATH
QADEA=/dr01/qadapps/qea
QADICT=/dr01/qadapps/ict
QADQDT=/dr01/qadapps/qdt
QADENV=live
PROPATH=$QADQDT/xmfqusrc:$QADQDT:$QADQDT/resources; export PROPATH

cd $QADQDT
$DLC/bin/_progres -pf $QADQDT/envs/$QADENV/scripts/batchCompile.pf -param
"rcodeDestFormat=1,compileListFilename=$QADICT/utcompil.wrk,destinationDirectory=
$QADICT,compilePropath=$QADICT/xrc:$QADEA/xrc:$QADEA/xrc/proxy:$QADEA/fin" >>
$QADICT/compileICT.log 2>&1

```

Run the above script and check the log file.

```

04/20/12 @ 10:04:54 [root] - Compiling /dr01/qadapps/ict/xrc/icuatxrp.p
04/20/12 @ 10:04:54 [root] - Compiling /dr01/qadapps/ict/xrc/icuipers.p
04/20/12 @ 10:04:54 [root] - Compiling /dr01/qadapps/ict/xrc/icuitmt.p
04/20/12 @ 10:04:54 [root] - Compiling /dr01/qadapps/ict/xrc/icvlmt.p
04/20/12 @ 10:04:54 [root] - Compiling /dr01/qadapps/ict/xrc/runProgramMapper.p
04/20/12 @ 10:04:54 [root] - Compiling /dr01/qadapps/ict/xrc/whapist.p
04/20/12 @ 10:04:55 [root] - Compiling /dr01/qadapps/ict/xrc/whapish.p
04/20/12 @ 10:04:55 [root] - Compiling /dr01/qadapps/qea/xrc/whgblmgr.p
04/20/12 @ 10:04:56 [root] - Compile contained 3 errors.
04/20/12 @ 10:04:56 [root] - Compile contained 0 warnings.

```

Make sure that the `Propath` is set properly to point to the `$QADICT` folder in the first position:

ICT,(NETUI),(QXTEND),(MSW/PSW),(WMS),CORE (all core layers, including BBI, FIN, QRA, PROXY)

Both the compilation and runtime `Propath` should be adjusted to the following order:

- 1 (CUST/PATCH/LOCAL)
- 2 ICT

- 3 (NETUI)
- 4 (QXTEND)
- 5 (MSW/PSW)
- 6 (WMS)
- 7 CORE (all core layers, including BBI, FIN, QRA, PROXY)

Enterprise Edition 2011 EE, 2011.1 EE, and 2012 EE

Compile ICT and EE modified programs using the QDT batch compiler solution.

UNIX

Create the `$QADICT/compileICT.ksh` script.

Make sure that the `Propath` and other parameters are properly set for an EE environment. Detailed information about how to run the QDT Batch Compiler is available in the existing batch compiler script—for example, `$QADQDT/envs/$QADENV/scripts/01batchCompile.ksh`—or in the `01batchCompile.log` files from the `$QADQDT/logs` directory.

The example below assumes you are installing ICT into `/dr01/qadapps/ict` to an existing 2012 EE installation found in `/dr01/qadapps/qea` with QDT in `/dr01/qadapps/qdt`.

```
DLC=/progress/dlc102b; export DLC
PATH=$PATH:$DLC/bin; export PATH
QADEA=/dr01/qadapps/qea
QADICT=/dr01/qadapps/ict
QADQDT=/dr01/qadapps/qdt
QADUI=$QADEA/qadui
QADQXTEND=$QADEA/qxtend
QADENV=live
PROPATH=$QADQDT/xmfgusrc:$QADQDT:$QADQDT/resources; export PROPATH
cd $QADQDT
$DLC/bin/_progres -pf $QADQDT/envs/$QADENV/scripts/batchCompile.pf -param
"rcodeDestFormat=1,compileListFilename=$QADICT/utcompil.wrk,destinationDirectory=
$QADICT,topLevelList=none,noCompileList=nocompile.lst,progOnlyList=
ProgressOnly.lst,oraOnlyList=OracleOnly.lst,srcCodeFormat=twoletter,compilePropath=
$QADICT/xrc:$QADUI:$QADQXTEND/xrc:$QADQXTEND:$QADEA/xrc:$QADEA/xrc/us/bbi:$QADEA/xrc/
proxy:$QADEA/fin" >> $QADICT/compileICT.log 2>&1
```

Check the compilation results in `$QADICT/compileICT.log`.

```
...
08/30/12 @ 16:07:27 [root] - Compiling /dr01/qadapps/ict/xrc/us/ic/ictstrp.p to
/dr01/qadapps/ict/us/ic
08/30/12 @ 16:07:28 [root] - Compiling /dr01/qadapps/ict/xrc/us/ic/ictuca.p to
/dr01/qadapps/ict/us/ic
08/30/12 @ 16:07:28 [root] - Compiling /dr01/qadapps/ict/xrc/us/ic/icuatrp.p to
/dr01/qadapps/ict/us/ic
08/30/12 @ 16:07:28 [root] - Compiling /dr01/qadapps/ict/xrc/us/ic/icuatxrp.p to
/dr01/qadapps/ict/us/ic
08/30/12 @ 16:07:29 [root] - Compiling /dr01/qadapps/ict/xrc/us/ic/icuipers.p to
/dr01/qadapps/ict/us/ic
08/30/12 @ 16:07:29 [root] - Compiling /dr01/qadapps/ict/xrc/us/ic/icuitmt.p to
/dr01/qadapps/ict/us/ic
08/30/12 @ 16:07:29 [root] - Compiling /dr01/qadapps/ict/xrc/us/ic/icvlmt.p to
/dr01/qadapps/ict/us/ic
08/30/12 @ 16:07:29 [root] - Compile contained 0 errors.
08/30/12 @ 16:07:29 [root] - Compile contained 0 warnings.
```

Windows

Create the %QADICT%/compileICT.bat script.

Make sure that the Propath and other parameters are properly set for an EE environment. Detailed information about how to run the QDT Batch Compiler is available in the existing batch compiler script—for example, %QADQDT%\envs\%QADENV%\scripts\01batchCompile.bat—or in the 01batchCompile.log files from the %QADQDT%\logs directory.

The example below assumes you are installing ICT into C:\ict to an existing 2012 EE installation found in C:\qadapps with QDT in C:\qdt.

```
SET DLC=C:\Progress\OpenEdge
SET PATH=%PATH%;%DLC%\bin
SET QADEA=\qadapps
SET QADICT=\ict
SET QADQDT=\qdt
SET QADUI=%QADEA%\qadui
SET QADQXTEND=%QADEA%\qxtend
SET QADENV=pilot
C:
cd %QADQDT%
C:\Progress\OpenEdge\bin\_progres.exe -pf C:\qdt\envs\pilot\scripts\batchCompile.pf -
ininame C:\qdt\ini\batchCompile.ini -param "rcodeDestFor-mat=1,compileListFilename=
%QADICT%\utcompil.wrk,destinationDirectory=%QADICT%,topLevelList=none,noCompileList=
nocompile.lst,progOnlyList=ProgressOnly.lst,oraOnlyList=OracleOnly.lst,srcCodeFormat=
twoletter,compilePropath=
%QADICT%\xrc;%QADUI%;%QADQXTEND%\xrc;%QADQXTEND%\tools;%QADEA%\xrc;%QADEA%\xrc\us\bbi;
%QADEA%\xrc\proxy;%QADEA%\fin" >> %QADICT%\compileICT.log 2>&1
```

Check the compilation results in %QADICT%\compileICT.log.

```
...
05/27/13 @ 07:32:29 [mfg] - Compiling \qadapps\ict\xrc\us\ic\ictstop.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:29 [mfg] - Compiling \qadapps\ict\xrc\us\ic\ictstrp.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:29 [mfg] - Compiling \qadapps\ict\xrc\us\ic\ictuca.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:29 [mfg] - Compiling \qadapps\ict\xrc\us\ic\icuatrp.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:29 [mfg] - Compiling \qadapps\ict\xrc\us\ic\icuatxrp.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:30 [mfg] - Compiling \qadapps\ict\xrc\us\ic\icuiipers.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:30 [mfg] - Compiling \qadapps\ict\xrc\us\ic\icuitmt.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:30 [mfg] - Compiling \qadapps\ict\xrc\us\ic\icvlmt.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:31 [mfg] - Compile contained 0 errors.
05/27/13 @ 07:32:31 [mfg] - Compile contained 0 warnings.
```

Propath Settings

Make sure that the Propath is set properly to point to the \$QADICT folder in the first position:

ICT,(NETUI),(QXTEND),(MSW/PSW),(WMS),CORE (all core layers, including BBI, FIN, QRA, PROXY)

Both the compilation and runtime Propath should be adjusted to the following order:

- 1 (CUST/PATCH/LOCAL)
- 2 ICT

- 3 (NETUI)
- 4 (QXTEND)
- 5 (MSW/PSW)
- 6 (WMS)
- 7 CORE (all core layers, including BBI, FIN, QRA, PROXY)

Enterprise Edition 2012.1 EE and Later

Compile ICT and EE modified programs using the QDT batch compiler solution.

UNIX

Create the `$QADICT/compileICT.ksh` script.

Make sure that the `Propath` and other parameters are properly set for an EE environment. Detailed information about how to run the QDT Batch Compiler is available in the existing batch compiler script—for example, `$QADQDT/envs/$QADENV/scripts/01batchCompile.ksh`—or in the `01batchCompile.log` files from the `$QADQDT/logs` directory.

The example below assumes that you are installing ICT into `/dr01/qadapps/ict` to an existing 2012.1 EE installation found in `/dr01/qadapps/qea` with QDT in `/dr01/qadapps/qdt`.

```
DLC=/progress/dlc102b; export DLC
PATH=$PATH:$DLC/bin; export PATH
QADEA=/dr01/qadapps/qea
QADICT=/dr01/qadapps/ict
QADQDT=/dr01/qadapps/qdt
QADUI=$QADEA/qadui
QADQXTEND=$QADEA/qxtend
QADENV=live
PROPATH=$QADQDT/xmfgsrc:$QADQDT:$QADQDT/resources; export PROPATH
cd $QADQDT
$DLC/bin/_progres -pf $QADQDT/envs/$QADENV/scripts/batchCompile.pf -param
"rcodeDestFormat=1,compileListFilename=$QADICT/utcompil.wrk,destinationDirectory=
$QADICT,topLevelList=none,noCompileList=nocompile.lst,progOnlyList=
ProgressOnly.lst,oraOnlyList=OracleOnly.lst,srcCodeFormat=twoletter,compilePropath=
$QADICT/xrc:$QADUI:$QADQXTEND/xrc:$QADQXTEND/tools:$QADEA/xrc:$QADEA/xrc/us/bbi:$QADEA
/xrc/ proxy:$QADEA/fin" >> $QADICT/compileICT.log 2>&1
```

Check the compilation results in `$QADICT/compileICT.log`.

```
...
08/30/12 @ 16:07:27 [root] - Compiling /dr01/qadapps/ict/xrc/us/ic/ictstrp.p to
/dr01/qadapps/ict/us/ic
08/30/12 @ 16:07:28 [root] - Compiling /dr01/qadapps/ict/xrc/us/ic/ictuca.p to
/dr01/qadapps/ict/us/ic
08/30/12 @ 16:07:28 [root] - Compiling /dr01/qadapps/ict/xrc/us/ic/icuatrp.p to
/dr01/qadapps/ict/us/ic
08/30/12 @ 16:07:28 [root] - Compiling /dr01/qadapps/ict/xrc/us/ic/icuatxrp.p to
/dr01/qadapps/ict/us/ic
08/30/12 @ 16:07:29 [root] - Compiling /dr01/qadapps/ict/xrc/us/ic/icuipers.p to
/dr01/qadapps/ict/us/ic
08/30/12 @ 16:07:29 [root] - Compiling /dr01/qadapps/ict/xrc/us/ic/icuitmt.p to
/dr01/qadapps/ict/us/ic
08/30/12 @ 16:07:29 [root] - Compiling /dr01/qadapps/ict/xrc/us/ic/icvlmt.p to
/dr01/qadapps/ict/us/ic
08/30/12 @ 16:07:29 [root] - Compile contained 0 errors.
08/30/12 @ 16:07:29 [root] - Compile contained 0 warnings.
```

Windows

Create the %QADICT%\compileICT.bat script.

Make sure that the Propath and other parameters are properly set for an EE environment. Detailed information about how to run the QDT Batch Compiler is available in the existing batch compiler script—for example, %QADQDT%\envs\%QADENV%\scripts\01batchCompile.bat—or in the 01batchCompile.log files from the %QADQDT%\logs directory.

The example below assumes that you are installing ICT into C:\ict to an existing 2012.1 EE installation found in C:\qadapps with QDT in C:\qdt.

```
SET DLC=C:\Progress\OpenEdge
SET PATH=%PATH%;%DLC%\bin
SET QADEA=\qadapps
SET QADICT=\ict
SET QADQDT=\qdt
SET QADUI=%QADEA%\qadui
SET QADQXTEND=%QADEA%\qxtend
SET QADENV=pilot
C:
cd %QADQDT%
C:\Progress\OpenEdge\bin\_progres.exe -pf C:\qdt\envs\pilot\scripts\batchCompile.pf -
ininame C:\qdt\ini\batchCompile.ini -param "rcodeDestFor-mat=1,compileListFilename=
%QADICT%\utcompil.wrk,destinationDirectory=%QADICT%,topLevelList=none,noCompileList=
nocompile.lst,progOnlyList=ProgressOnly.lst,oraOnlyList=OracleOnly.lst,srcCodeFormat=
twoletter,compilePropath=
%QADICT%\xrc;%QADUI%;%QADQXTEND%\xrc;%QADQXTEND%\tools;%QADEA%\xrc;%QADEA%\xrc\us\bbi;
%QADEA%\xrc\proxy;%QADEA%\fin" >> %QADICT%\compileICT.log 2>&1
```

Check the compilation results in %QADICT%\compileICT.log.

```
...
05/27/13 @ 07:32:29 [mfg] - Compiling \qadapps\ict\xrc\us\ic\ictstop.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:29 [mfg] - Compiling \qadapps\ict\xrc\us\ic\ictstrp.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:29 [mfg] - Compiling \qadapps\ict\xrc\us\ic\ictuca.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:29 [mfg] - Compiling \qadapps\ict\xrc\us\ic\icuatrp.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:29 [mfg] - Compiling \qadapps\ict\xrc\us\ic\icuatxrp.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:30 [mfg] - Compiling \qadapps\ict\xrc\us\ic\icuiipers.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:30 [mfg] - Compiling \qadapps\ict\xrc\us\ic\icuitmt.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:30 [mfg] - Compiling \qadapps\ict\xrc\us\ic\icvlmt.p to
\qadapps\ict\us\ic
05/27/13 @ 07:32:31 [mfg] - Compile contained 0 errors.
05/27/13 @ 07:32:31 [mfg] - Compile contained 0 warnings.
```

Propath Settings

Make sure that the Propath is set properly to point to the \$QADICT folder in the first position:

ICT,(NETUI),(QXTEND),(MSW/PSW),(WMS),CORE (all core layers, including BBI, FIN,
QRA, PROXY),(CONFIGURATOR),(CRM)

Both the compilation and runtime Propath should be adjusted to the following order:

- 1 (CUST/PATCH/LOCAL)
- 2 ICT

- 3 (NETUI)
- 4 (QXTEND)
- 5 (MSW/PSW)
- 6 (WMS)
- 7 CORE (all core layers, including BBI, FIN, QRA, PROXY)
- 8 (CONFIGURATOR)
- 9 (CRM)

Additional Tasks for All Versions

After compilation, perform the tasks described in this section for all versions of QAD Enterprise Applications.

Make sure that your ERP application processes (including connection manager, telnet, and character client) are running with the OS user ID that belongs to the same OS group, to ensure that the newly created program files will be accessible from both .NET UI and character UI.

For this reason, the umask for these users should be set to 002 instead of the default 022. This value can be set either in a user login profile (.profile, .bash_profile, .bashrc, and so on, depending on the default shell assigned to the user) or in the application startup scripts (connmgr.*, telnet.*, client.*)

Update Runtime Scripts and Properties Files

Add \$QADICT to the Propath for all QAD Enterprise Applications processes, including character client, connection manager, telnet, appserver, webspeed.

The \$QADICT directory should be added just after the PATCH or CUST layer, before other QAD products folders, to follow the order: (CUST/PATCH/LOCAL), ICT, (NETUI), (QXTEND), (MSW/PSW), (WMS), CORE (all core layers, including BBI, FIN, QRA, PROXY).

Update the QGen client to run \$QADICT/runProgramMapper .r.

Start QAD Enterprise Applications

If no compilation errors occurred, log on to the environment for which you installed QAD ICT. At this point, QAD ICT is not active yet. QAD ICT data (such as message numbers and menus) must be loaded and QAD ICT must be activated.

Note If menu 90 has already been used, you can manually update the qadict.dmp. For any other change, please contact QAD Services.

Note The qadict.dmp is provided for the iso8859-1 and compatible codepages setup. If you run an incompatible codepage setup (for example, 1250), the qadict.dmp must be adjusted accordingly:

```
[Session]
Numeric-format=AMERICAN
Date-Format=mdy
```

```
cpstream=1250
cpinternal=1250
```

UPGRADE NOTE QAD ICT 1.0 has a different menu structure. Because of this, if you upgrade ICT 1.0 to ICT 4.0 or higher, the existing ICT menu must be deleted before you start loading data. To delete the whole QAD ICT menu, run the `icmenrm.p` program.

```
+-----+
|Warning!!Procedure will delete the existing Q&D ICT menu. ||Start deleting: yes|
+-----+
```

UPGRADE NOTE If you upgrade QAD Enterprise Application Enterprise Edition, you must perform a System Sync with the Resource option selected using System Synchronize after running `icmenrm.p`.

UPGRADE NOTE If you upgrade QAD Enterprise Applications from a non-domained version, you must run the conversion tool `ictcnv36.p` to add the proper domain into your customizations. Run the tool after you load all customizations into the new system. After logon, start the QAD ICT Data Loader program `icdatmt.p` from the menu. For an Enterprise Edition release, go to the editor (via `mgeditor`) and execute the 4GL command: `run us/ic/icdatmt.p` (as the menu permissions have not been set yet).

In the loader program, fill in the QAD ICT dump file name `qadict.dmp`. (The `qadict.dmp` supports both Standard Edition and Enterprise Edition.)

```
icdatmt.p ict4          90.24.2 Load Developers Data          04/12/13
                Load Procedure for Customisation
-----
Dump file:
Active Trig: Yes          Preview processed data: No
Overwrite: Yes           UNDO/delete data on DMP: No
  QAD ICT parameters: No
    Menus: No
    Window help: No
  Generalized codes: No
    Messages: No
  Language details: No
    Browsers: No
  Web-enabled programs: No
  Other customisations: No
```

WARNING: Continue with dump/load. Yes

Enter data or press F4 to end.

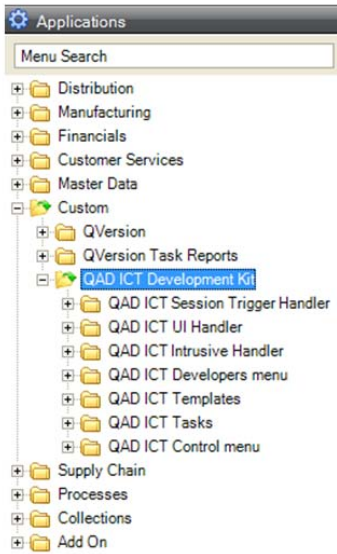
After confirmation, QAD ICT loader will load the data and summarize what has been loaded (details can also be found in the log file). The number of records can differ depending on which QAD ICT release you are installing.

```
Total number of records: 1935
          Added: 1934
    Overwritten: 1
      Skipped: 0
      Deleted: 0

see logfile for more detailed infomation: load01.log
```

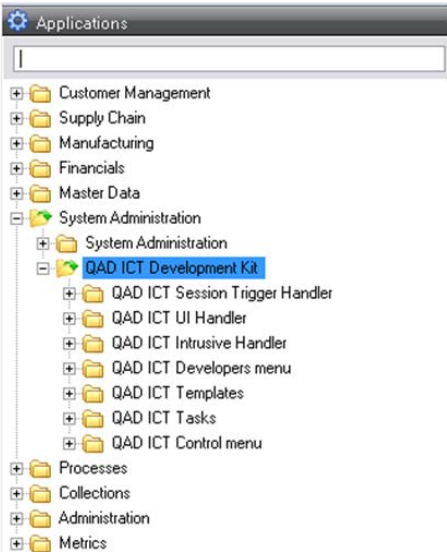
When installing in a Standard Edition release, you should have the QAD ICT menu available as option 90 in the main system menu (if it was not changed manually in the `qadict.dmp` file).

In .NET UI, you will see the ICT menu in the Custom menu, QAD ICT Development Kit.



When installing in an Enterprise Edition release, synchronization should be performed automatically during `.dmp` file loading. If you received any synchronization errors when loading the `.dmp` file, perform a System Sync with the Resource option selected using System Synchronize. Do a log-off/on and run Role Permissions Maintain (for SuperUser) and update permissions to ICT menu options accordingly. See the QAD Enterprise Edition *Security and Controls User Guide* for information about setting up security roles.

In .NET UI, after log-off/on, you will see QAD ICT in the System Administration Menu, QAD ICT Development Kit.



Define QAD ICT Licensing

Run License Registration (36.16.10.1) and register the license provided by GCA. See the QAD Enterprise Application user guide section on License Registration for details.

To make sure that QAD ICT will be loaded during startup, run Super Program Maintenance (90.24.13) and check for three records. The most important one is `icprcs.p`.

```
icspmt.p ict4 a          90.24.13 Super Program Maintenance          03/31/12
```

Super Program: icprcs.p	
Comment:	Active: Yes Primary: Yes

If the Active and Primary fields are not set to Yes, QAD ICT might not work correctly as the QAD ICT package must be loaded (during startup) as the first (super) package.

In addition to `icprcs.p`, also make sure that `icrctrg.p` (for Active change Triggers) is defined as a super procedure and that Active is set to Yes and Primary to No.

Set Up QAD ICT Control Table

QAD ICT Control Table (90.24.24) defines where the ICT Developer Menu is installed. If during the initial load of QAD ICT data the menu numbers were changed, enter here the changed menu number in this program.

The language prefix defines the default language that is being used for the customer (in most cases, that will be US).

Task base directory is used to indicate where task subdirectories are created. Task directory is obtained by concatenating the base directory and the task ID. The new or modified custom procedures are kept in the task directory and compiled there as long as the task is open. Once the task is closed, all such files are moved automatically to the central directory (r-codes to `global_user_lang_dir + first two letters subdirectory`, source code files to the first source subdirectory) and thus become visible to all users.

Central directory is used to indicate the directory where the customer customizations are stored (subscriber, UI triggered, validation, other non ICT-related procedures). This directory is usually the first one on the Propath.

Both the task base directory and central directory must have full write, read, and execute rights for further ICT developers.

Source sub-dir defines a comma-separated list of source subdirectories used to extend the Propath. The first subdirectory is the target subdirectory for customized source files in the central directory.

```

icctpm.p ict4 a          90.24.24 ICT Control Table          08/29/12

Active Change triggers: Yes
                        ICT Menu: 90.12
ICT Menu Accelerator CTRL-: R
Trigger link file prefix: ic  trgXX
Audit for ICT qad_wkfl: No
Language prefix: us

Task base directory: /u1/qad/ea2010se/cust/ict-tasks
Central directory: /u1/qad/ea2010se/cust
Source sub-dir.: src,xrc

Reload QAD ICT: No      ICT VERSION: 4.1
    
```

QAD ICT is reloaded if 'Reload QAD ICT' flag is set to Yes and F1(go) was pressed. Optionally, you can log off and log on again to QAD Enterprise Applications. The ICT processing is restarted ONLY for the current user. For .NET UI/Desktop users, restart the Connection Manager.

UPGRADE NOTE When you upgrade your QAD ICT installation, it is necessary to run two procedures that automatically adjust your existing customizations to ICT 4.1 requirements. It is very important to run them after ICT Control Table configuration. Remember to close all WIP ICT tasks; otherwise, ICT closes them automatically. Run the program `ictcnv35.p` to update the existing Frame & Fields settings. Run the utility `ictcnv34.p` (it is best to run the compiled version, `us/ic/ictcnv34.r`) to convert existing ICT tasks to conform to the new version of ICT. This procedure assigns all existing customizations that are not linked with any task to the new created task. (The system prompts you for the task name during the conversion.)

UPGRADE NOTE When you upgrade your QAD ICT installation, it is important to authorize all developed customizations. To do this, run the authorization tool program `icauthmt.p`. Execute `icauthmt.p` and reload QADICT.

Validate that QAD ICT Is Working

Key 1	Key 2	Key 3	Key 4	Character Field 1	Decimal Field 1
QadIctPersistentProg icacco.p			1046	activateTriggers us/	0.00
QadIctPersistentProg icentron.p			7445	startUIEntry us/ic/i	0.00
QadIctPersistentProg ictrg00.p			7443	activateTriggers us/	0.00
QadIctPersistentProg icuipers.p			7444	activateTriggers us/	0.00
QadIctSuperProcedure icprocs.p			7439	us/ic/ictstart.p	0.00
QadIctSuperProcedure icctrq.p			7441	us/ic/ictstart.p	0.00
Trigger	delete brw_mstr	us/ic/icrctrq.p	?	us/ic/icrctrq.p	0.00

If you do not get this screen (search for Key 1 data QadIctSuperProcedure) or you get run-time error messages (for example, *** "getTempTable" was not found. (293)), QAD ICT is not loaded (via the restart or log off/on).

If this happens, make sure that the Propath is set up correctly and is first pointing to the customized QAD ICT `mf1.p`. If you cannot resolve the issue, contact QAD Services.

Run the same validation in .NET UI. The same information as in Character UI should be displayed.

If QXtend Inbound is installed:

- 1 Run the 90.24.10 report in a QGen session. This should result in the following information:

```

ictstrp.p ict4 a 90.24.10 Reporting
Page: 1 MFG/PRO Training DB - eB2.1

Temp ICT records
-----
Key 1 Key 2 Key 3 Key 4
-----
QadIctPersistentProg icentron.p 1013
QadIctSuperProcedure icacc.p 1011
QadIctSuperProcedure icprcs.p 1009
QadIctSuperProcedure icrcrtrg.p 1012

Trigger create brw_mstr us/ic/icrcrtrg.p ?
Trigger create code_mstr us/ic/icrcrtrg.p ?
Trigger create cref_mstr us/ic/icrcrtrg.p ?
Trigger create drl_mstr us/ic/icrcrtrg.p ?
Trigger create flh_mstr us/ic/icrcrtrg.p ?

Browse or type search string and press [RETURN]:
Use cursor and page keys to navigate, F4 to exit. [Line 1]
    
```

- 2 Run the QXtend UI adapter script. (The exact name and path of this script can be found in the QXtend UI Adapter connection pool configuration settings; for example, /u1/qad/ea2010se/qxtend/client.qxtend.)

Enter or paste the following connection string in the empty field displayed by the UI adapter script (it requires that the “mfg” user in the application has no password defined):

app=ictstrp.p;id=qps;ip=127.0.0.1;term=vt320;xenguser=mfg:

and run the report with Output to “terminal” instead of “page.”

ict4 ict4 a

```

Session ICT settings: No
Defined ICT settings: No

Temp ICT records: Yes

Output: terminal
Batch ID:
    
```

```

ict4 a Date: 03/31/12 qad.int
Page: 1 Time: 21:48:17

Temp ICT records
-----
Key 1 Key 2 Key 3 Key 4 Character Field 1 Decimal Field 1
-----
QadIctPersistentProg icentron.p 1045 startUIEntry us/ic/i 0.00
QadIctPersistentProg icuipers.p 1044 activateTriggers us/ 0.00
QadIctSuperProcedure icacc.p 1042 us/ic/icmf.p 0.00
QadIctSuperProcedure icprcs.p 1040 us/ic/icmf.p 0.00
QadIctSuperProcedure icrcrtrg.p 1043 us/ic/icmf.p 0.00
    
```

Important Note that procedure `icacc.p` in the above screenshots was used before ICT 4.0. Since ICT 4.0, it has been replaced by `icaccx.p` (where `x` is the shortcut key) or is not present at all (if the shortcut key is defined in ICT Control Table, 90.24.24).