Your Business ▪ Your Agenda ▪ Our Passion
Maximizing the value of IT together

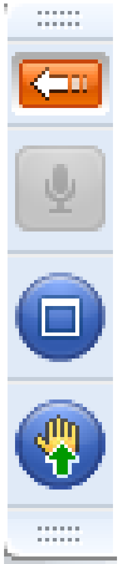
Certification Preparation for U2 Developers

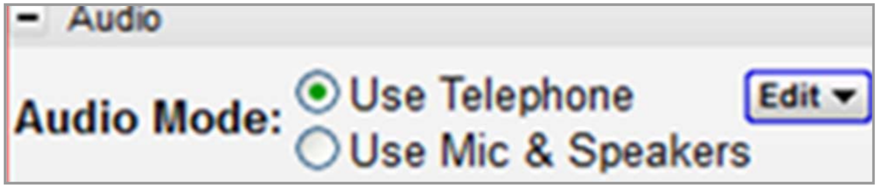
Michael C. Ryan
Senior Technical Instructor

Rocket U2™

Webinar Opening Procedure

- Orange arrow = control GoTo Webinar control panel
- This webinar will be recorded and posted to our website for you to replay at a later date
  - You will be notified via email once .pdf and video are posted
- If using phone – don’t select "Use Mic & Speakers"

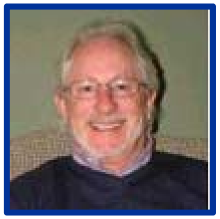




− Audio

Audio Mode: ● Use Telephone ○ Use Mic & Speakers

- All attendees are muted during the presentation
- Brief Q&A session – after the presentation

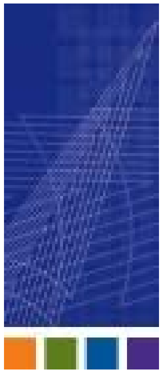


Michael C. Ryan’s Biography

- Joined UniData in 1994 as a Support Engineer with a three-year assignment to Sydney, Australia
- Current role as U2 Senior Technical Instructor
- Responsible for creating, writing, and presenting U2 (UniData and UniVerse) courseware worldwide for U2 Education
- Expertise across the product line including tools such as SB/XA, U2 Web DE and U2 DataVu
- Over 20 years experience with MultiValue databases
- Over 15 years experience in the education role

Abstract: Certification Preparation for U2 Developers

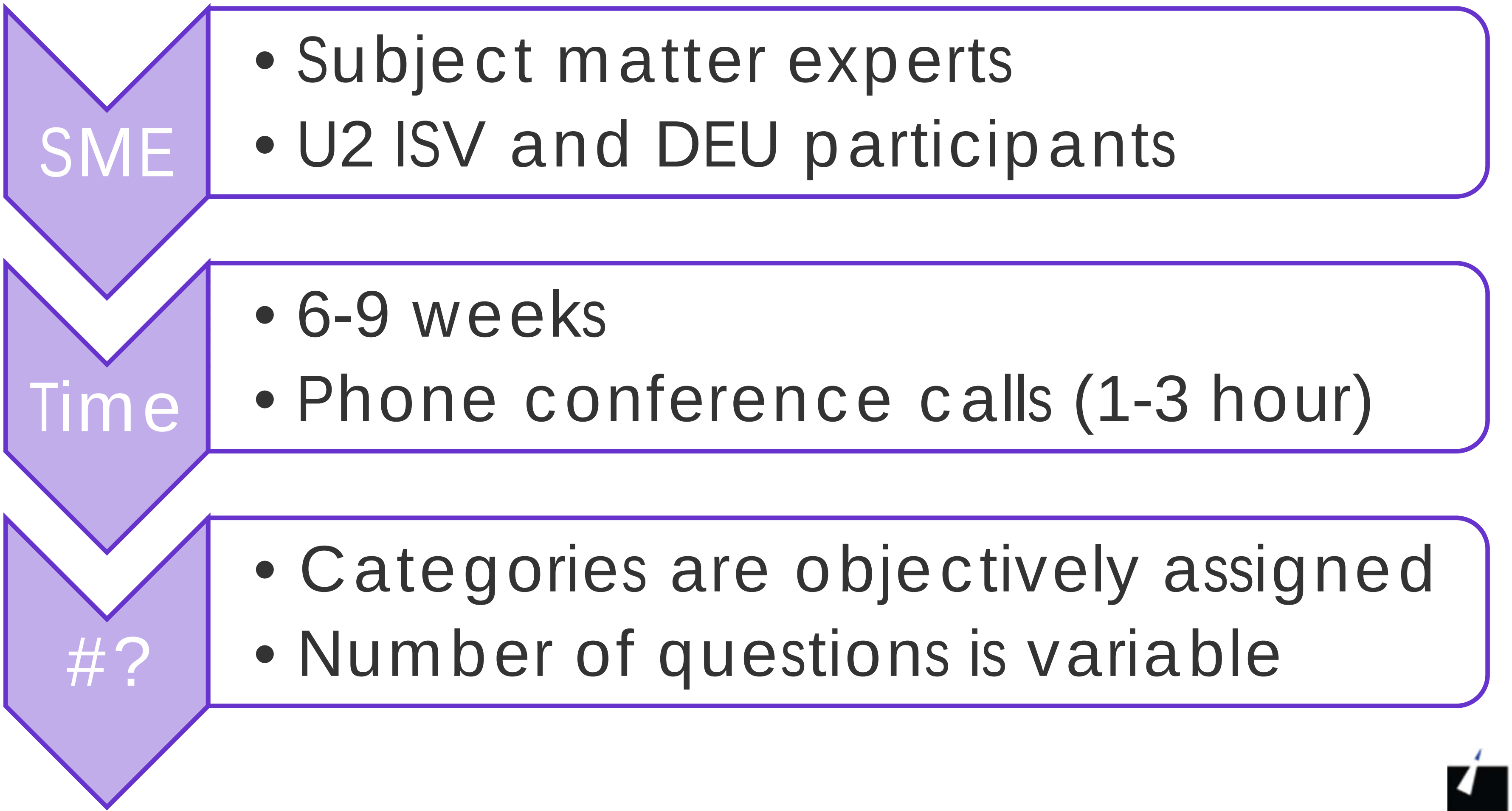
This certification prep course will help you prepare to take the U2 Developer certification exam by reviewing the topics that will be covered on the exam. In addition to answering questions specific functional areas.



Agenda

- How exams are created
- Blueprint of the exam
- What you need to know
- Investigate topics and possibilities
- Education programs available
- Q&A session after presentation

Who Creates the Exam?



SME

- Subject matter experts
- U2 ISV and DEU participants

Time

- 6-9 weeks
- Phone conference calls (1-3 hour)

#?

- Categories are objectively assigned
- Number of questions is variable

Programmer Tasks

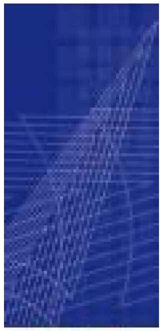
Task Domain Covered

1. Application design
2. Logic constructs
3. User interface
4. File handling
5. Data formatting and manipulation
6. Array, variable and compiler directives
7. System variables and functions
8. Deployment
9. Debugging programs
10. BASIC extensions

The Exam

- Exam A, B, or Practice
- 90 minutes per exam
- Can take twice in one day
- Passing score is 67%

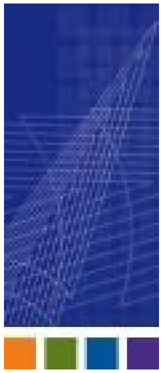
Section Title	Number of Items (65)
Application design	4
Logic constructs, statement label handling	6
User interface	7
File handling	14
Data formatting and manipulation	9
Arrays, variables & computer directives	9
System variables and functions	7
Deployment	3
Debugging	3
BASIC extensions	3



Topics Review

Programmer Responsibilities

1. Application design
2. Logic constructs
3. User interface
4. File handling
5. Data formatting and manipulation
6. Array, variable and compiler directives
7. System variables and functions
8. Deployment
9. Debugging programs
10. BASIC extensions



Section 1 - Application Design

- Create hashed files
- Define dictionaries
- Define indices
- Integrate application tools
  - Paragraphs
  - UniQuery (UD)
  - RetrieVe (UV)
  - SQL
- Use External subroutines

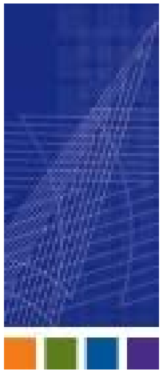
Pseudo Code Example:

```
If student's grade is
greater than or equal
to 60

   Print "passed"

   or else

   Print "failed"
```



Section 1 – Application Design (cont'd)

- Is there a dictionary type designed specifically for "user use"?
- What is a BASIC statement?
- Is it mandatory to produce pseudo code before writing code?
- Is it necessary to have a file dictionary built before writing code?
- Is indexing important to the application?

Section 1 – Application Design *(cont’d)*

- Name the two types of subroutines
- Are there specific keywords necessary in every subroutine?
- How many arguments can be passed in a subroutine?
- What must occur after a successful compilation of a subroutine in order to use it?

Section 2 – Logic Constructs

- Should you always have an END statement following an IF?
- What is the rule regarding END statements in a program?
- What statement can you use that helps align programs for readability and understanding (in edit session)?

Section 2 – Line Constructs *(cont'd)*

Explain code segment – objective is to print CERT NOW

```
A=1; B=2; C=3

CRT "Line 1"

IF (A=B) THEN IF (B=C) THEN CRT
"CERT Later" ELSE PRINT "CERT NOW"

CRT "Line 2"

IF (A=B) THEN IF (B=C) THEN CRT
"CERT Later" ELSE PRINT "What?" ELSE
PRINT "CERT NOW"

CRT "Line 3"

IF (A>B) THEN IF (B>C) THEN CRT
"CERT Later" ELSE PRINT "CERT NOW"

CRT "Line 4"

IF (A<B) THEN IF (B<C) THEN CRT
"CERT Later" ELSE PRINT "CERT NOW"
```

ANSWER

Line 1
Line 2
CERT NOW
Line 3
Line 4
CERT Later

Section 2 – Line Constructs *(cont'd)*

- What is the purpose of the MATCHES statement?
- Is there a difference in MATCHES and !MATCHES?
- Determine the correct MATCH for the following to evoke an @TRUE value:
  - 123456789
  - 349-xx-U2FUN
- Name the match operators
- What is the significance of the "~" in a MATCH statement?

Section 2 – Line Constructs *(cont'd)*

- Comparison operators produce what result?
- Can you use the Query language operators EQ, NE, LT, GT, LE GE in a program?
- List the four ways you could produce "not equal“ a comparison result
- In the following statement, name the order of execution:

```
If A=12; B=3; C=2; D=5

result = A + B / C * D
```

… and what is the value of result?

Section 2 – Line Constructs *(cont'd)*

- Which keyword allows you to use a functional I-Type or Virtual dictionary item in a program?
- Can you define your own functions? If so, which keyword would you use?
- What is the purpose of the Statement Label?
- Discuss the rules for using a Statement Label.
- Can you use GOTO to move to the following statement label?

```
1000%3: READ REC FROM F.VAR, 123 ELSE…
```

Section 3 – User Interface

- Where would the statement "Greetings" be located on a typical 80 X 23 screen using: CRT @(0,70)

- Explain the difference of the CRT command and the PRINT command

- Explain the following code and the reason for its use:

```
CRT @(-1)
VAR = "STU DENT010"
MAX = LEN(VAR)
CRT "CONV" : "   " : "VAR"
FOR X = 1 TO MAX
  CRT SEQ(VAR[X,1]) :"   ": VAR[X,1]
NEXT I
```

Section 3 – User Interface *(cont'd)*

- Explain ICONV and OCONV
- Explain the ICONV codes below:

D	Date
MC	Masked character
MD	Masked decimal
MT	Masked time
P	Pattern
R	Range

Section 3 – User Interface *(cont’d)*

```
TODAY = DATE()
CRT TODAY
TD = TIMEDATE()
THIRTYBACK = TODAY - 30
CRT TD
CRT "Thirty days ago..." ; CRT
THIRTYBACK
CRT OCONV(TODAY,"D2/")
CRT OCONV(TODAY,"DWADMAY")
CRT OCONV(TODAY,"D4 JY")
CRT OCONV(TODAY,"D2 YA")
CRT OCONV(TODAY,"D MR")
CRT OCONV(TODAY,"D D")
CRT OCONV(TODAY,"DMDYL[A,'th,',4]")
CRT OCONV( TODAY, "D-DMY[2,2,2]")
```

```
15592
07:53:21 08 SEP 2010
Thirty days ago...     15562
09/08/10
WEDNESDAY 08 SEPTEMBER 2010
251 2010
TIGER
IX
08
September 08th, 2010
08-09-10
```

Section 3 – User Interface *(cont'd)*

- Given: internal time of day is 31654

.... which converted is 08:47
.... which converted is 08:47:34
.... which converted is 08:47AM

- Which OCONV time code conversion are used in the above code segment?

Section 3 – User Interface *(cont'd)*

Masked character output conversion:

- Which conversion code will return only alpha characters from a string?
  - NON alpha?
- Which conversion code will convert upper case letters to lower case letters?
  - The inverse?
  - Title case?
- Which conversion code will return only numeric characters from a string?

Section 3 – User Interface *(cont'd)*

Masked decimal output conversion:

- Given 1234567 which conversion code will produce the following output?

    - `12345.67`
    - `12,345.67`
    - `$12345.67`
    - `$12345.67db`
    - `<12345.67>`

Section 3 – User Interface *(cont'd)*

- Formatting code snippet:

```
PRINT "Name ": (user enters Samantha)
INPUT NAME
NAME = FMT(NAME,"L#3")
IF NAME = "Samantha" THEN NAME = "SAMANTHA"
PRINT "Name is ":NAME
```

- What is the output?

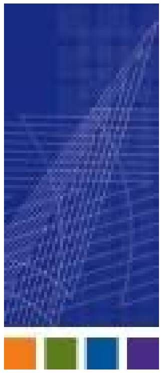
Section 4 – File Handling

- When dealing with U2 files, four clauses are available as generic I/O. In which order MUST these clauses be used?

```
THEN
LOCKED
ELSE
ON ERROR
```

Section 4 – File Handling *(cont'd)*

- What is the purpose of the ON ERROR clause?
- What is the purpose of the LOCKED clause?
- When would the THEN clause be invoked?
- When would the ELSE clause be invoked?

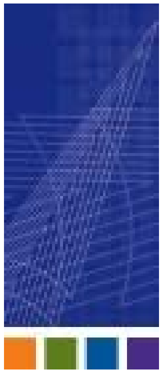


Section 4 – File Handling *(cont’d)*

- Given the code snippet:

```
 OPEN "DICT", "INVENTORY" TO INV.FVAR
 ON ERROR
    CRT "CANT OPEN THE FILE"
    STOP
 END
 ELSE
* No problems found
 END
. . . . . .
```

- What is the effect of the above code?



Section 4 – File Handling *(cont’d)*

- What must occur before using any of the following functions?

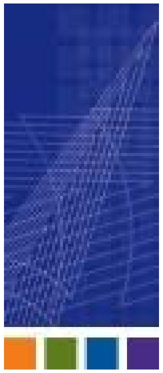
```
INMAT()
FILEINFO()
INDICES()
```

- If you wanted to read a line of a sequential text file, which command would you use?

Section 4 – File Handling *(cont'd)*

- To write to just one field on a record you could use which command?
- Which command will set a single lock that locks every record in a file?
- When using the SELECT f.var statement in a program is it permissible to select a dynamic array?
- In what order are records returned after using the SELECT f.var command?
- What are two effects of using the RELEASE statement?

Section 5 – Data Formatting and Manipulation



- Given: `"Zellars, Dan 1 Camel Back Suite 3300 London"`

  Using a "delimited" extract field, complete the command that will populate the following variables?

```
FirstName = FIELD()  ; CRT FirstName

LastName = FIELD()   ; CRT LastName

OfficeAddr = FIELD(); CRT OfficeAddr

City = FIELD()        ; CRT City
```

Section 5 – Data Formatting and Manipulation *(cont’d)*

- Explain the purpose of the below functions:

```
COUNT()
DCOUNT()
INDEX()
LEN()
```

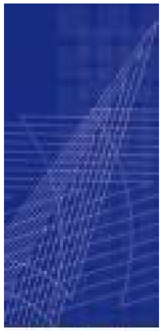
Section 5 – Data Formatting and Manipulation *(cont'd)*

- What is the output of the following code snippet?

```
DMY.STRING = FIELD(TIMEDATE()," ",2,3)
CRT "Entire Date = ":QUOTE(TIMEDATE())
CRT "Date Portion = ":QUOTE(DMY.STRING)
CRT "COL1() = ":COL1()
CRT "COL2() = ":COL2()
```

```
Entire Date = "08:59:37 Sep 09 2010"
Date Portion = "Sep 09 2010"
COL1() = 9
COL2() = 21
```

Section 5 – Data Formatting and Manipulation *(cont'd)*



- Explain the difference between the two functions:

  `SUBR()` `ITYPE()`

- Which command would you use to determine:
  - If a file is open
  - Return the VOC name for the file
  - Path to the file
  - Type of file (static, dynamic)
  - If the file has an index

Section 6 – Arrays and Variables

- What are the type of arrays supported in U2?
- Name the array hierarchy.
- Explain the purpose of the array assignment:
    - Extraction
    - Replacement
    - Insertion
    - Deletion

Section 6 – Arrays and Variables

(cont'd)

- Explain the difference in the two array commands:

  ```
  FIND
  FINDSTR
  ```

- When updating a multivalued associated field, does the U2 BASIC perform an "associated update" automatically?

Section 6 – Arrays and Variables
(cont'd)

- Explain the results of the following three array arithmetic operations:

1

```
QTYS = 14 : @VM : 20 : @VM : 5
COST = 2.5 : @VM : 3.6 : @VM : 12
CRT QTYS * COST
```

2

```
QTYS = 14 : @VM : 20 : @VM : 5
COST = 2.5
CRT QTYS * COST
```

3

```
QTYS = 14 : @VM : 20 : @VM : 5
COST = 2.5
CRT QTYS * REUSE(COST)
```

Section 6 – Arrays and Variables

(cont'd)

- What is the output of the following code:

```
VAL="Now is the perfect time for Tea."
FORMAT = "6T"
FORMATTED = FMT(VAL,FORMAT)
LINES = DCOUNT(FORMATTED, @TM)
FOR I = 1 TO LINES
  CRT FIELD(FORMATTED, @TM,I)
NEXT I
```

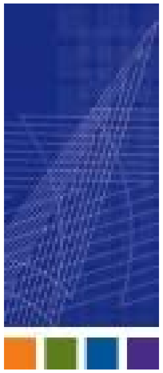
- What is printed on the fourth line?

Section 7 – System Variables

- Explain the difference between
  - Common
  - Named common
- Explain the purpose of compiler directives
- Explain the `EQUATE` statement
- Explain the purpose of `INCLUDE`

Section 7 – System Variables

(cont’d)



- Provide the system variable for:

```
"Login Name           = "
"User Number          = "
"Internal Date        = "
"This month is        = "
"Today is             = "
"This year is         = "
"Last command is      = "
"Last sentence is     = "
"Account Path is      = "
"Select Active        = "
"Account Name         = "
```

Section 8 - Deployment

- Explain "catalog" and type:

TYPE	UNIDATA	UNIVERSE
Local	X	X
Normal		X
Direct	X	X
Global	X	X

Section 8 – Deployment *(cont'd)*

- Explain the value of phantom application deployment.
- Can application programs be executed by phantom from the TCL/ECL prompt?
- When a parent of a phantom process has an untimely death, what happens to the child process?

Section 9 - Debugging

- When launching a program from TCL/ECL what is the option to invoke the debugger immediately?
- While in a debug session (from TCL/ECL) which switch command will display code about to be executed?
- Which GUI based debugging tools can you employ to debug application programs?

Section 10 – BASIC Extensions

- Which two Eclipse-based tools that are used to assist programmers in creating XML and Web services?
- Where are XML documents, maps, DTDs and schema items are stored
(UV and UD)
- Is the default U2 XML output element or attribute centric?

Section 10 – BASIC Extensions
(cont’d)

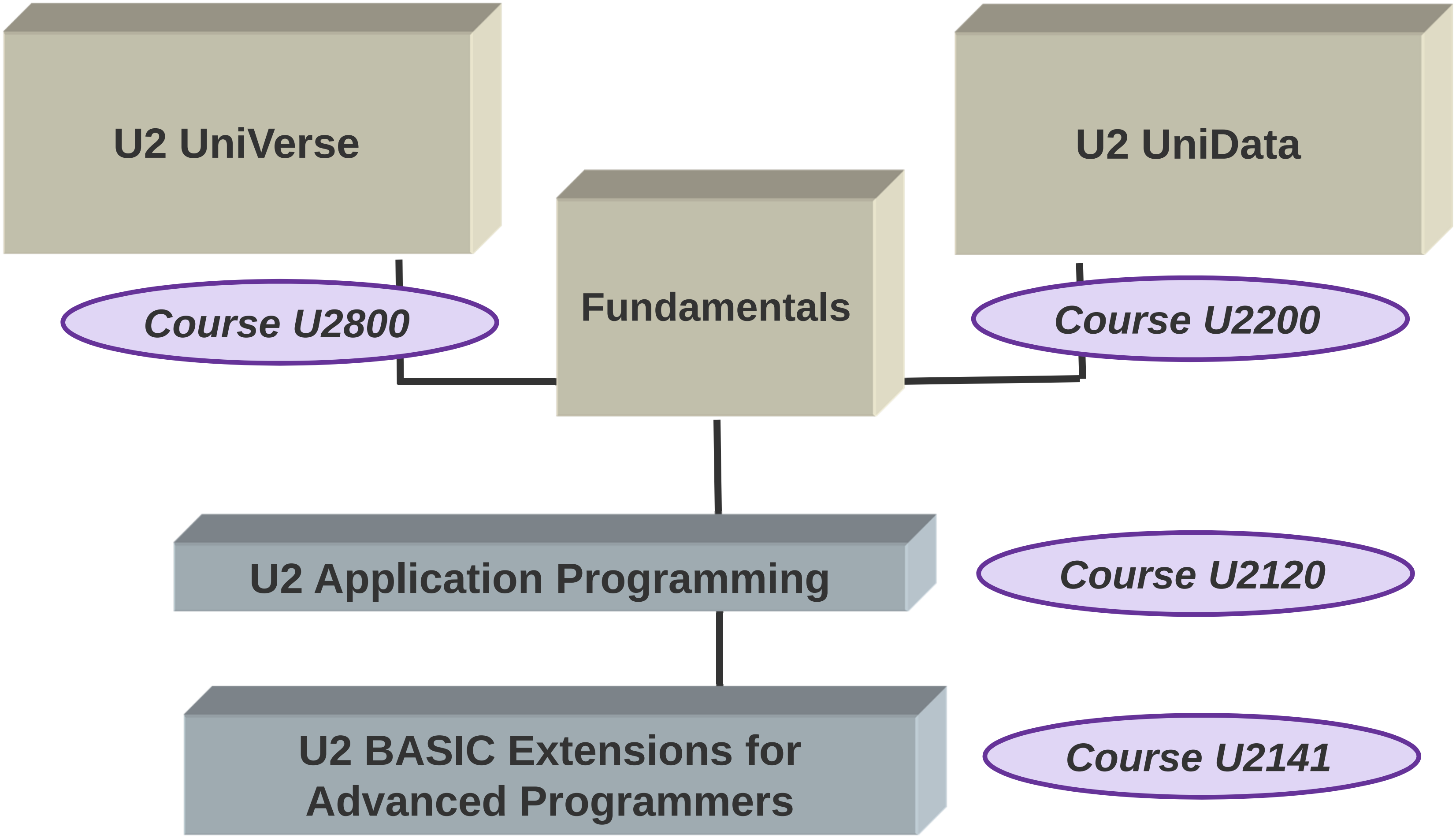
- Explain the value of reading and writing to a socket.
- Which BASIC function will translate an XML document to a HTML document based on a map?
- What is the default transport protocol for the CallHTTP ( ) function?

Your roadmap to success...

Application Developer

Rocket U2™

U2 Application Developer Education Roadmap



Additional Resources

- Developing UniBasic Applications (UV and UD documentation manuals)
- New Features documentation
- U2 TechConnect

For more information see: www.rocketsoftware.com/u2/training

Exams are **FREE** at U2 University!

- Exams are offered during all three days of U2 University 2012 in all three venues
- Check out our agenda for times at: http://u2u.rocketsoftware.com

Summary

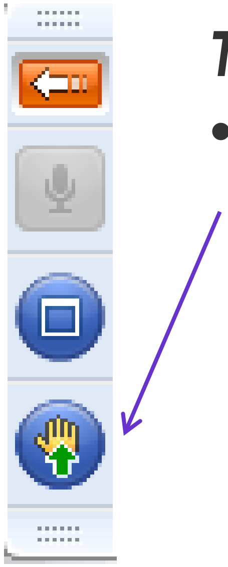
How exams are created

Investigated topics and possibilities

Education programs available

How you can prepare

Question and Answer Session



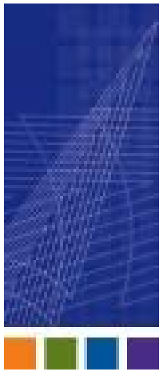
To ask a question:

- Click on hand icon with green arrow and we will call your name

Or you may email us your question later at:

U2Training@rocketsoftware.com

Disclaimers



THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

IN ADDITION, THIS INFORMATION IS BASED ON ROCKET SOFTWARE'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY ROCKET SOFTWAREWITHOUT NOTICE.

ROCKET SOFTWARE SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM ROCKET SOFTWARE (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF ROCKET SOFTWARESOFTWARE.

Trademarks and Acknowledgments

The following are trademarks or registered trademarks of Rocket Software, Inc.: UniData, wIntegrate and Dynamic Connect.

IBM, the IBM logo, AIX, DB2 and the DB2 Universal Database are trademarks of IBM in the United States and other countries.

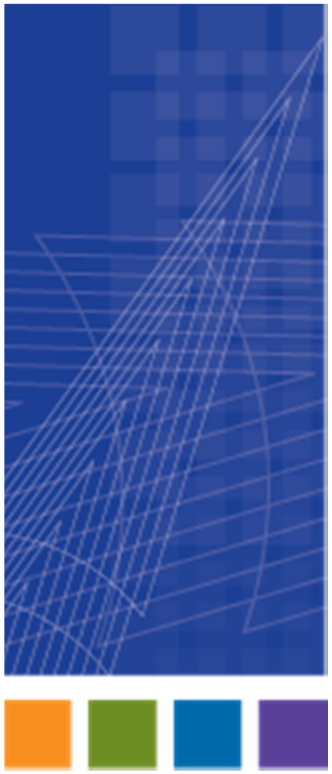
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Microsoft, Windows, Windows NT, and Excel are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names mentioned herein may be trademarks or service marks of others.

Thank
You
Gracias
Dank u wel
Mahalo
Juspaxar
Khawp khun
Spasibo
Efcharisto
Salamat
Merci
Toda raba
Cám ơn
Dhanyawaad
Köszi
Grazie mille
Kamsahamnida
Pidamayado
Dankon
Tusen takk
Terima kasih
Shukuria
Tashakkur
bolzïn
Paldies
Maake
Tingki
Komapsumnida
gozaimashita
atu
Tesekkurler
Ekhmet
Mehrbani
Arigato
Dankscheen
Doh je
suksama
Yaqhanyelay
Efcharisto
Shukria
Biyan
Obrigado
Dhanyabaad
Yuspagrasunki
Merastawhy



Thank you!