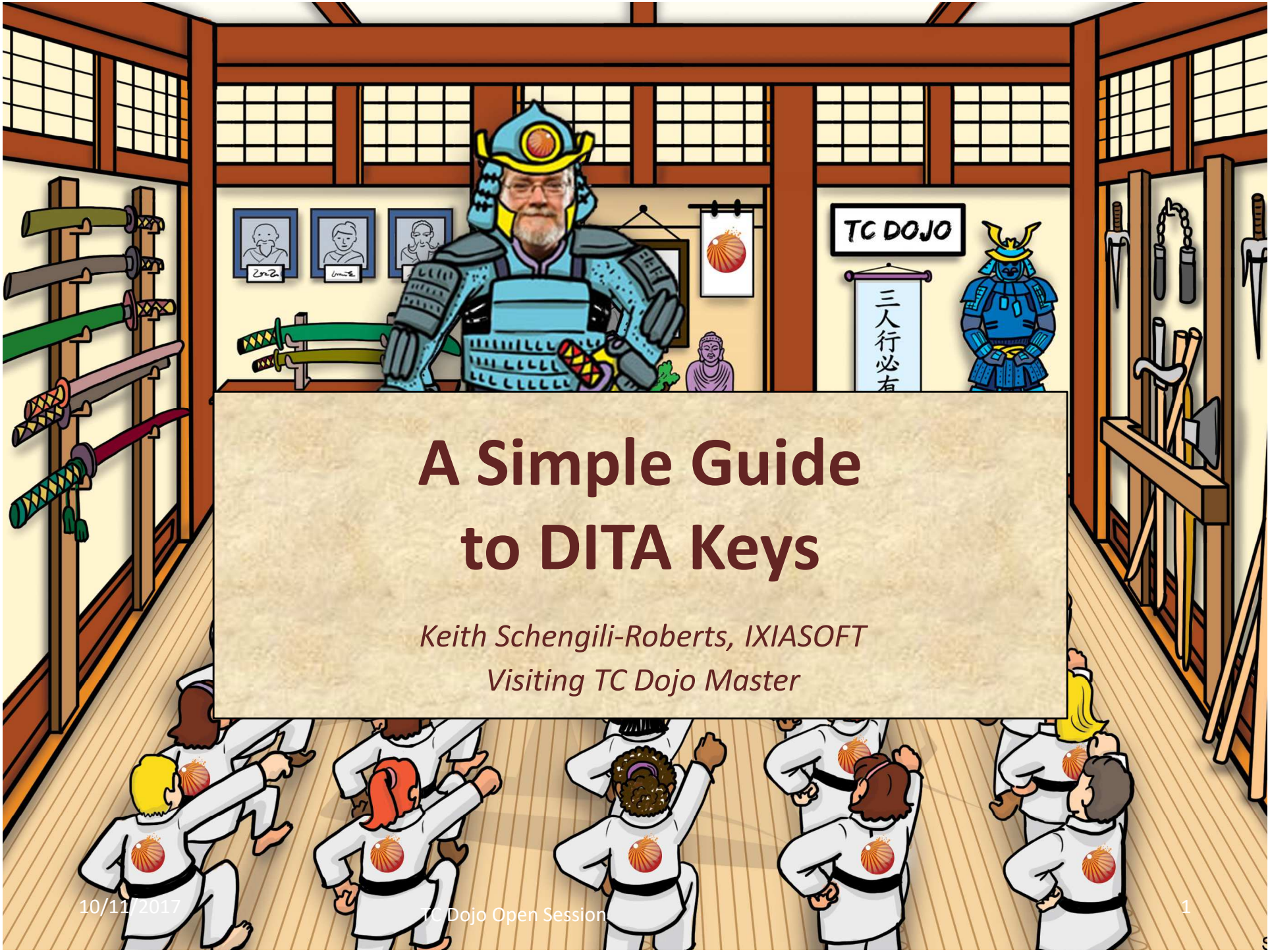


A Simple Guide to DITA Keys

Keith Schengili-Roberts, IXIASOFT
Visiting TC Dojo Master

Agenda

- Introduction
- The How and Why of DITA Keys
- What are the Advantages of Keys
- Using Keys to Handle Document “Variables”
- Using Keys to Reference Images
- Creating an “Image Warehouse” Adding Keys to Topicrefs
- Keys and Conditional Processing
- How Far Down the Rabbit Hole Do You Go?
- Other DITA Key Best Practices
- Further Reading/Viewing
- Q/A

Who's This Guy?

Keith Schengili-Roberts, DITA Evangelist and Market Researcher with IXIASOFT

What I do:

- Advocate for DITA use and talk about best practices
- Liaison with OASIS; Chair of the DITA Adoption Committee, and member of the Lightweight DITA and DITA Technical Committees
- Industry researcher
- Have 12+ Years of experience with DITA XML

Am Also "DITAWriter"

- Industry blog started +6 years ago
- Almost 275,000 hits
- Have regularly updated info on DITA Conferences, DITA Books, Companies Using DITA, DITA CMSes, DITA Editors, other DITA Tools, and DITA Consulting Firms
- News and views on DITA use
- Also features interviews with those making a difference in the world of DITA

How this Webinar Came About

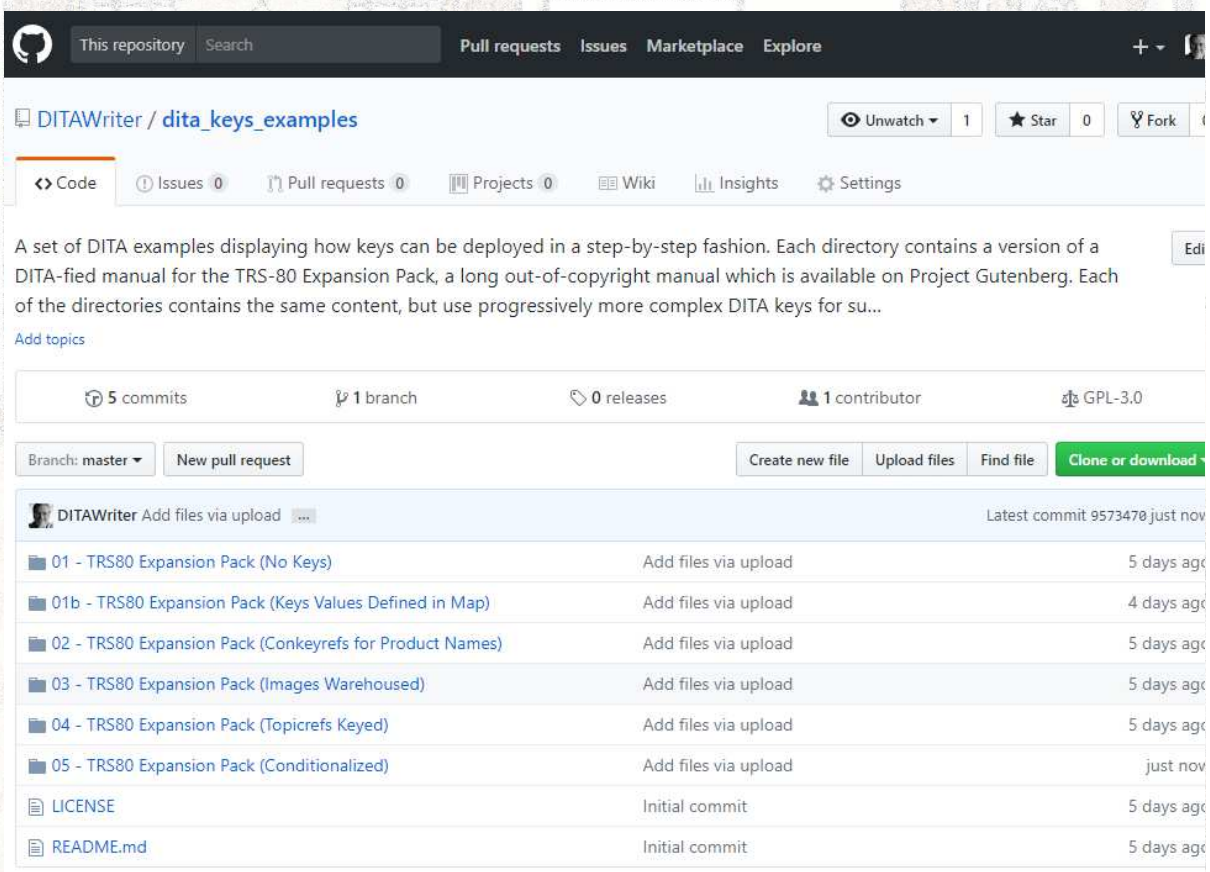
- Was at the STC conference earlier this year, and I talked to a couple of people on the opening day who liked the idea of using DITA, but had trouble understanding the concept of keys
- Liz was at the same conference, and I asked her if I could tackle the subject of DITA keys, with an emphasis on the basics

Taking a Different Tack

- There is too often a focus on the mechanics of keys, with very little discussion as to the “why”
- I plan to answer not only “how does it work?”, but also “why would you want to do that?”
- This webinar looks at the *basics*, using step-by-step code samples that build upon each other
- While this presentation will use screenshots of the oXygen XML editor, focus is on code rather than tools

Code Samples!

- I will be using code examples throughout this webinar
- They show a step-by-step, progression (and abstraction) of key use
- The code examples can be downloaded from github.com/DITAWriter/dita_keys_examples



Target Document: TRS-80 Expansion Interface

- Delving into the past of computing history, I am using the manual for a peripheral device for the venerable "Trash 80" computer from the late 1970s/early 1980s
- It is out of copyright and is freely available from Project Gutenberg at https://www.gutenberg.org/ebooks/27469
- I have "DITA-fied" it and it is the basis for the samples used in this webinar

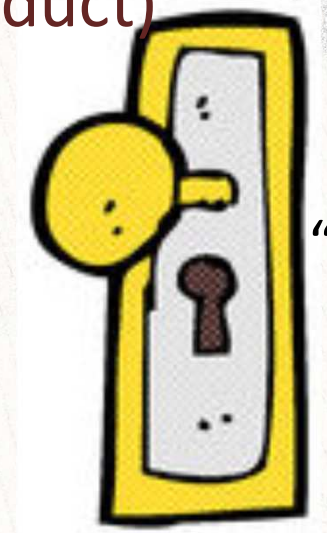
The How and Why of DITA Keys

So What Are Keys Really?

- Keys are an indirect referencing mechanism
- What do we mean by “indirect”?
  - Think of this way: instead of pointing directly to a thing such as an external URL, variable text strings, images, etc., you use a key whose name can be referenced instead

Direct addressing: “TRS-80” (name of product)

Indirect addressing: 


“TRS-80”

Indirect Referencing Has a Long History

- The idea of indirect referencing within computer science goes back to the mid-1960s
- SNOBOL (StriNg Oriented and symBOlic Language) used it back when the IBM 7094 was the computing king

[173]
SNOBOL

SNOBOL

Early scientific applications of computers were dominated by numerical computation. As the enormous potential of accurate, high-speed calculation became evident, attention turned to nonnumerical applications: the manipulation of text and structured relationships among objects. Applications such as machine translation of languages, text concordances, and manipulation of algebraic formulas motivated the development of programming languages in which nonnumeric computations could be specified. SNOBOL (StriNg Oriented symBOlic Language) developed in this environment. While most recently developed programming languages have some facilities for manipulation of nonnumerical objects, SNOBOL stands out for its high level text processing capabilities. As a result it is in wide use for applications involving nonnumeric processing.

SNOBOL has developed over a period of time from a primitive text processor to a powerful general-purpose language still emphasizing text manipulation. The present version of the language is SNOBOL4, subsequently referred to here simply as SNOBOL.

[179]
SNOBOL

Indirect Referencing

While it is common for low-level machine languages to have an operation that specifies a data object indirectly by means of a remote address, such an operation is rarely found in high-level languages. SNOBOL is the major exception.

In SNOBOL, the prefix operator $ causes the value of its operand, rather than the operand itself, to be used as an identifier. For example, if

```
VOLUME = "LOUD"
```

Then

```
$VOLUME = 17
```

is equivalent to

```
LOUD = 17
```

This indirect referencing, which can be applied to any number of levels, is useful for constructing associative relationships among objects and for computing transfers on the basis of data values not known when the program is constructed. For example the goto :($ROUTE) transfers to the label that is the value of ROUTE, not to the

Maps Provide the Context for Keys

- This is the main difference between conrefs and keys; as keys are always “scoped” to the map that they reside in
- If context is for Kings, then DITA maps are definitely royalty ;)

What are the Advantage of Keys?

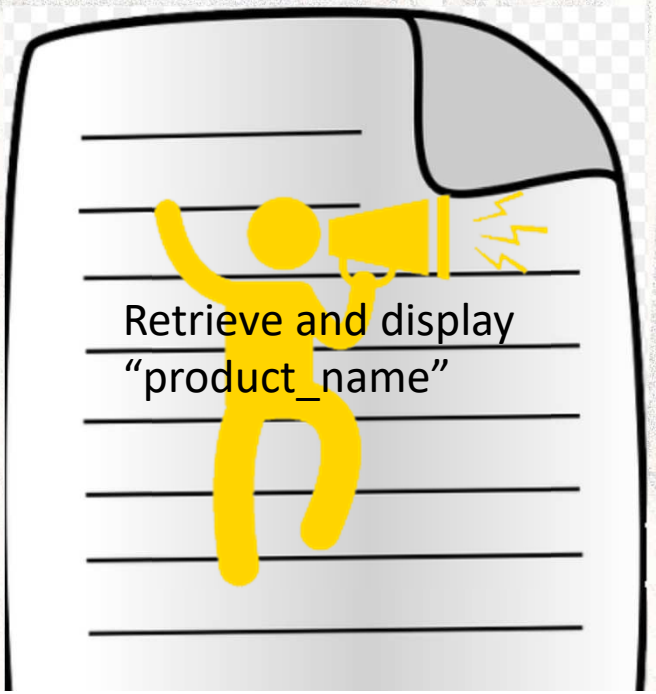
- You can reference the key name for a thing, and it will use the value assigned to it
- Instead of changing the name for a product, URL, image, etc., in multiple places, you only need to change it once in the parent map
- You can then use the key name anywhere in your topics and it will be displayed ("resolved") in place
  - Can also use conditional processing to change which key is referenced in a topic or which key definition is "active" in the map
- This is useful to ensure consistent naming and terminology across a document set
  - Built-in consistency!
  - Content reuse!

The Parts of a Key

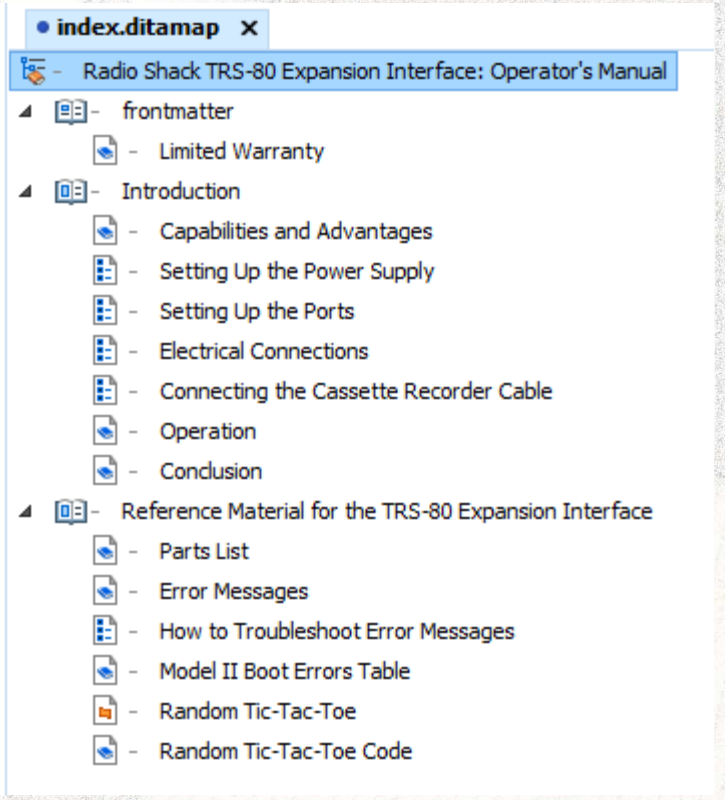
Keys always consist of at least three parts:

1. Declared key “name” for a thing (key)
2. The target for that thing (key definition or key value)
3. Calling upon the key name when needed (key reference)

“TRS-80”



An Initial, No Keys View of the Bookma



Bookmap view from oXygen's DITA Maps Manager

index.ditamap

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE bookmap PUBLIC "-//OASIS//DTD DITA BookMap//EN" "bookmap.dtd">
<bookmap>
    <booktitle>
        <mainbooktitle>Radio Shack TRS-80 Expansion Interface: Operator's Manual</mainbooktitle>
        <booktitlealt>Operator's Manual Catalog Numbers: 26-1140, 26-1141, 26-1142</booktitlealt>
    </booktitle>
    <bookmeta>
        <searchtitle>A DITA-fied Example of TRS-80 Expansion Interface Manual</searchtitle>
        <shortdesc>This manual for a long-obsolete computer peripheral is designed to be an example
            of how to port legacy content to DITA. </shortdesc>
        <author type="creator">Anonymous</author>
        <bookrights>
            <bookowner>
                <organization>Project Gutenberg</organization>
            </bookowner>
        </bookrights>
    </bookmeta>
    <frontmatter>
        <topicref href="topics/limited_warranty.dita"/>
    </frontmatter>
    <part href="topics/introduction.dita">
        <topicref href="topics/capabilities_and_advantages.dita"/>
        <topicref href="topics/setting_up.dita"/>
        <topicref href="topics/setting_up_ports.dita"/>
        <topicref href="topics/electrical_connections.dita"/>
        <topicref href="topics/connecting_cassette_recorder.dita"/>
        <topicref href="topics/operation.dita"/>
        <topicref href="topics/conclusion.dita"/>
    </part>
    <part href="topics/reference_material.dita">
        <topicref href="topics/parts.dita"/>
        <topicref href="topics/error_messages.dita"/>
        <topicref href="topics/error_messages_procedure.dita"/>
        <topicref href="topics/model_II_boot_errors_table.dita"/>
        <topicref href="topics/random_tic-tac-toe.dita"/>
        <topicref href="topics/random_tic-tac-toe_code.dita"/>
    </part>
</bookmap>
```

A code-level view of the same map

Using Keys to Handle Document "Variables"

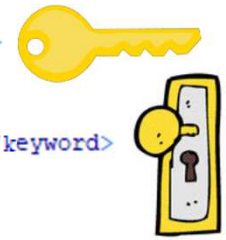
- It is handy to have "variables" (or in DITA, the key-based equivalent) to handle one- or two-word terms that are prone to change in different contexts, such as:
  - Company Name
  - Product Name
  - Model Number
  - Trademarked or Copyrighted terms, etc.
- Keys ensure that these terms are correctly used throughout all technical documents

Defining Keys Directly in the Map

- Simplest case is to define keys within the publication map using keydef + keyword

bookmap

```
<part href="topics/introduction.dita">
    <keydef keys="company_name">
        <topicmeta>
            <keywords>
                <keyword>Radio Shack</keyword>
            </keywords>
        </topicmeta>
    </keydef>
    <keydef keys="computer_name">
        <topicmeta>
            <keywords>
                <keyword>TRS-80</keyword>
            </keywords>
        </topicmeta>
    </keydef>
    <keydef keys="peripheral_name">
        <topicmeta>
            <keywords>
                <keyword>Expansion Interface</keyword>
            </keywords>
        </topicmeta>
    </keydef>
```



Within a topic referenced from the map

```
<p>The Screen Printer and Line Printer allow you to obtain hard copy (print
    information generated by your <ph keyref="computer_name"/>.</p>
```

  - Problem with this approach is that it limits reuse of the key definitions because you have to define them in each map.

Using a DITA Topic to Store Key Values for the Map

- The next level of key abstraction is to instead point to a DITA topic from the map that contains these “variables”
- This uses keydef, which is a topicref for keys
- topicgroup processing-role is set to “resource-only”, which means it will not appear directly in output
- keydef = key definition
- keys = the name location being declared
- In this case it points to a .dita file containing the product names

The Content of that Keyed .dita File

Product Variables

Short Description: Variable definitions that are used throughout the documentation

Variable names

colspecs...

Used as...	Name to use
Company name	Radio Shack **Comment:** Change this value should name of the company change.
Computer Name	TRS-80 **Comment:** Change this value when the name for the computer changes.
Peripheral Name	Expansion Interface **Comment:** Change this value should name of the peripheral change.

Full topic as seen using oXygen's Author view

The portion of the code referencing the computer name

```
<row>
  <entry>
    <p><ph>Computer Name</ph></p>
  </entry>
  <entry>
    <p><ph id="computer_name">TRS-80</ph></p>
    <draft-comment>Change this value when the name for the computer
      changes.</draft-comment>
  </entry>
</row>
```

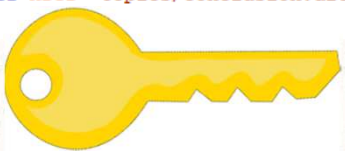
- The key targets are phrase tags using specific ids (id="computer_name") surrounding the value it represents ("TRS-80")

Using conkeyref to Call the Key Value

- conkeyref (content key reference, or “conref using a key”) is used to call the keyed value
- Uses two parts: keys name + id of specific key

```
<step>
    <cmd>Now connect the Ribbon Cable between the left front <ph
        conkeyref="product_info/peripheral_name"/> port and the <ph
        conkeyref="product_info/computer_name"/> port.</cmd>
</step>
```

```
<frontmatter>
    <topicref href="topics/limited_warranty.dita"/>
</frontmatter>
<part href="topics/introduction.dita">
    <topicgroup processing-role="resource-only">
        <keydef keys="product_info" href="topics/product_info.dita"/>
    </topicgroup>
    <topicref href="topics/capabilities_and_advantages.dita"/>
    <topicref href="topics/setting_up.dita"/>
    <topicref href="topics/setting_up_ports.dita"/>
    <topicref href="topics/electrical_connections.dita"/>
    <topicref href="topics/connecting_cassette_recorder.dita"/>
    <topicref href="topics/operation.dita"/>
    <topicref href="topics/conclusion.dita"/>
</part>
```



```
<row>
  <entry>
    <p><ph>Computer Name</ph></p>
  </entry>
  <entry>
    <p><ph id="computer_name">TRS-80</ph></p>
    <draft-comment>Change this value when the name for the computer
      changes.</draft-comment>
  </entry>
</row>
```

How the Keys are Resolved

- Author view in oXygen for this topic, showing the resolved key values
- Important that the tag used in the id matches that of the target; for inline text <ph> works well

Electrical Connections

Short Description: How to set up the electrical connections for peripheral devices attaching to the ▹TRS-80◃ computer.

1. ▹Turn the ▹TRS-80◃ so that it faces away from you. Locate the port Door (1400083); it's at the right end of the rear panel.◃

2. ▹To remove the Door, raise it up and slide it to the right—then lift it up and away from the ▹TRS-80◃.◃

3. ▹Place the ▹TRS-80◃ and ▹Expansion Interface◃ Hoods (14000217 and 14000214) on the Ribbon Cable Connectors as shown in Figure 4. The Hoods replace the Door on the ▹TRS-80◃ and fill the opening on the ▹Expansion Interface◃. These Hoods are designed so that it is not possible to insert the connectors upside down. They function as keyways for the connectors.◃

4. ▹Now connect the Ribbon Cable between the left front ▹Expansion Interface◃ port and the ▹TRS-80◃ port.◃

5. ▹Connect the DC Power Cord (DIN connector) to the POWER connector on the right rear of the ▹TRS-80◃ and connect both AC Power Cords to standard 120 VAC outlets.◃

But why stop with product names/model numbers?

Using Keys to Reference Images

- This document uses several images, which could be used in other publications

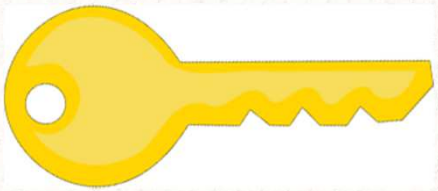
```
<fig>
    <title>Rear View—Interface Connections.</title>
    <image href="../images/figure_5.png"/>
</fig>
<fig>
    <title>Placement of <ph conkeyref="product_info/peripheral_name"/>.</title>
    <image href="../images/figure_6.jpg"/>
</fig>
```

- We can use keys to reference the images, so that they can be easily reused in subsequent publications

Creating an “Image Warehouse”

New keydef added to map pointing to image_warehouse.dita

```
<part href="topics/introduction.dita">
    <topicgroup processing-role="resource-only">
        <keydef keys="product_info" href="topics/product_info.dita"/>
        <keydef keys="image_warehouse" href="topics/image_warehouse.dita"/>
    </topicgroup>
    <topicref href="topics/capabilities_and_advantages.dita"/>
    <topicref href="topics/setting_up.dita"/>
    <topicref href="topics/setting_up_ports.dita"/>
    <topicref href="topics/electrical_connections.dita"/>
    <topicref href="topics/connecting_cassette_recorder.dita"/>
    <topicref href="topics/operation.dita"/>
    <topicref href="topics/conclusion.dita"/>
</part>
```



Two sample images with id values within image_warehouse.dita

```
<row>
    <entry><image id="fig5" href="../images/figure_5.png"/></entry>
    <entry>Rear View—Interface Connections</entry>
</row>
<row>
    <entry><image id="fig6" href="../images/figure_6.jpg"/></entry>
    <entry>Placement of <ph conkeyref="product_info/peripheral_name"
        /></entry>
</row>
```

Recommended Best Practice for Warehoused Content: Use Tables

- Makes it easy to tell at a glance what the image and its associated id is for calling them within a conkeyref
- Same goes for the product info, external URLs (e.g. company website, technical support), that are frequently referenced in your publications

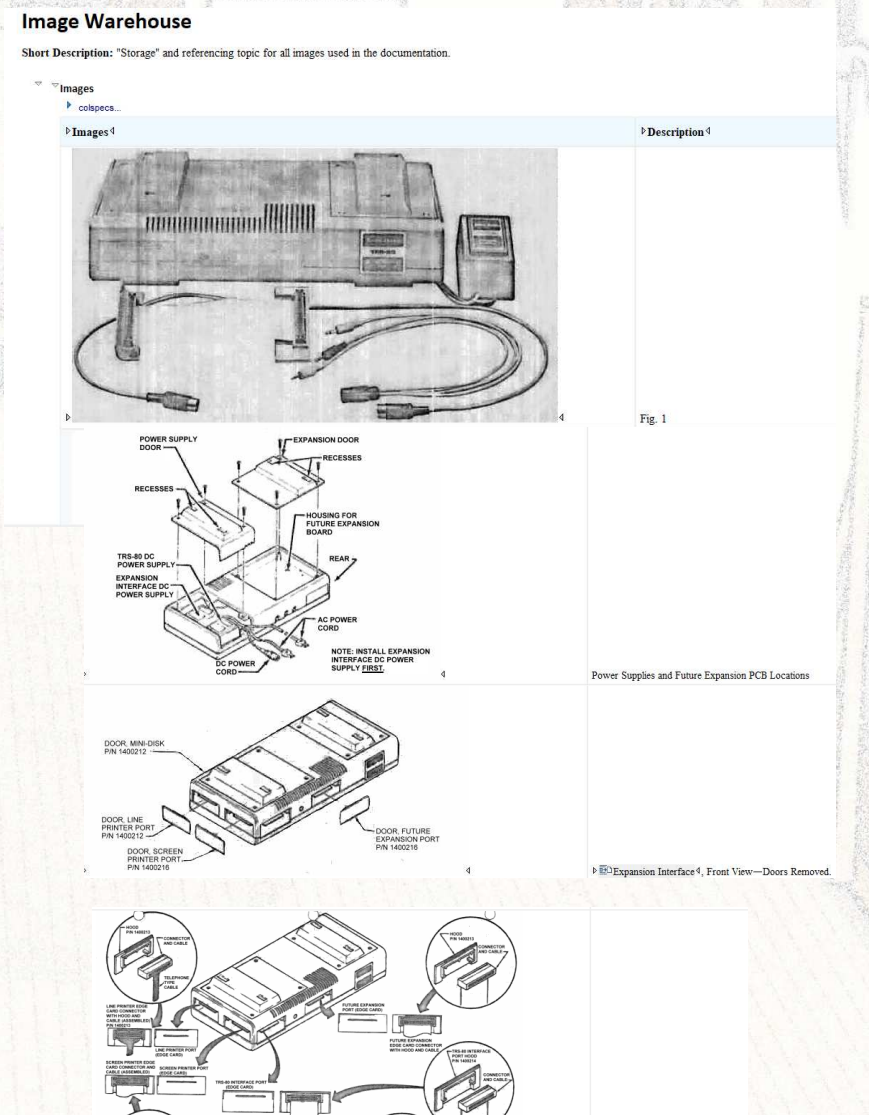


Image Warehouse

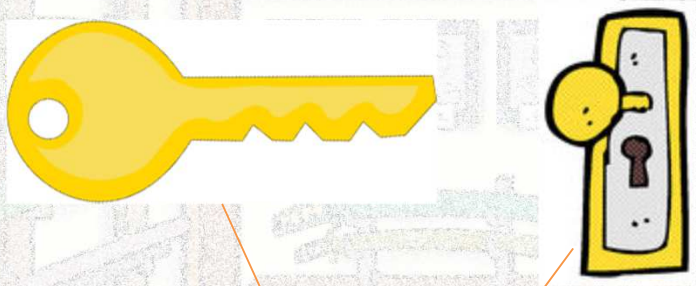
Short Description: "Storage" and referencing topic for all images used in the documentation.

Images

colspecs...

Images	Description
	Fig. 1
	Power Supplies and Future Expansion PCB Locations
	Expansion Interface, Front View—Doors Removed.

Calling an Image Using Conkeyref



```
<fig>
    <title>Rear View—Interface Connections.</title>
    <image conkeyref="image_warehouse/fig5"/>
</fig>
<fig>
    <title>Placement of <ph conkeyref="product_info/peripheral_name"/>.</title>
    <image conkeyref="image_warehouse/fig6"/>
</fig>
```

Conclusion

Short Description: Additional information, including block diagram, interface connections, and the suggested placement for the expansic interface.

Syntax:

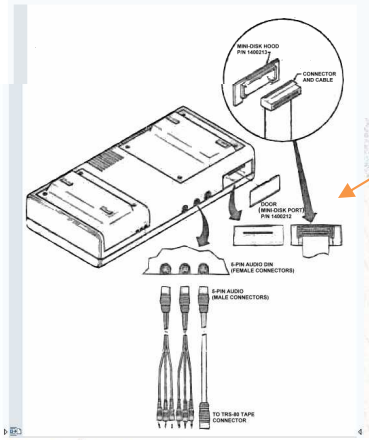
Possibly, you will not need all of the expansion modules that are available but, we have supplied you with Hoods for cable connectors for a complete expansion system. Use the Hoods as illustrated to prevent accidental mismatch between the edge connectors on the PCB and the cable connectors.

In the event that you lose a Door or Hood and want to replace it, we have given you a Parts List. You may refer to the Parts List and exploded diagrams to determine its Part Number. You can order replacement parts through your local TRS-80 Radio Shack store.

You must have a LEVEL II BASIC TRS-80 Microcomputer to utilize the TRS-80 Expansion Interface, the Line Printer and the Mini-Disk modules. If you have a LEVEL I BASIC machine, it must be modified to accept LEVEL II programs. The Screen Printer is the only expansion module that may be connected directly to the TRS-80 Microcomputer and that will operate with LEVEL I machines.

We are continually improving and updating our TRS-80 Microcomputer System. You will be kept informed through our Newsletters (you are on the mailing list), addenda and revisions to the Manual.

Rear View—Interface Connections.

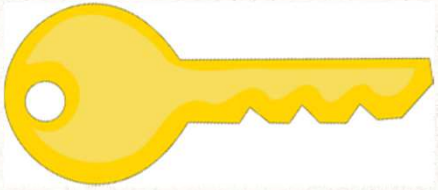


Placement of TRS-80 Expansion Interface.

But Wait, Why Stop There?

- To further increase topic level reuse between documents, can add key names to your topicrefs
- These can be referenced via a separate map file that can "warehouse" all commonly-used topics

```
<part keys="Introduction" keyref="introduction">
    <mapref href="topics/keydefs_topics.ditamap"/>
    <topicgroup processing-role="resource-only">
        <keydef keys="product_info" href="topics/product_info.dita"/>
        <keydef keys="image_warehouse" href="topics/image_warehouse.dita"/
    </topicgroup>
```



New mapref added to bookmap pointing to keydefs_topics.dita

```
<!DOCTYPE map PUBLIC "-//OASIS//DTD DITA Map//EN" "map.dtd">
<map>
  <title>Key definitions for all content topics</title>
  <topicmeta>
    <shortdesc>These topics have a corresponding key definition which can be used from the parent
      bookmap. These resource-only key definitions can be used for content references (@conkeyref)
      or from navigation topicrefs.</shortdesc>
  </topicmeta>
  <topicgroup>
    <keydef keys="capabilities_and_advantages" href="capabilities_and_advantages.dita"/>
    <keydef keys="conclusion" href="conclusion.dita"/>
    <keydef keys="connecting_cassette_recorder" href="connecting_cassette_recorder.dita"/>
    <keydef keys="electrical_connections" href="electrical_connections.dita"/>
    <keydef keys="error_messages" href="error_messages.dita"/>
    <keydef keys="error_messages_procedure" href="error_messages_procedure.dita"/>
    <keydef keys="introduction" href="introduction.dita"/>
    <keydef keys="limited_warranty" href="limited_warranty.dita"/>
    <keydef keys="model_II_boot_errors_table" href="model_II_boot_errors_table.dita"/>
    <keydef keys="operation" href="operation.dita"/>
    <keydef keys="parts" href="parts.dita"/>
    <keydef keys="random_tic-tac-toe" href="random_tic-tac-toe.dita"/>
    <keydef keys="random_tic-tac-toe_code" href="random_tic-tac-toe_code.dita"/>
    <keydef keys="reference_material" href="reference_material.dita"/>
    <keydef keys="setting_up" href="setting_up.dita"/>
    <keydef keys="setting_up_ports" href="setting_up_ports.dita"/>
  </topicgroup>
</map>
```

Contents of keydefs_topics.dita, which uses @keys to name individual topics

Adding Keys to Topicrefs

- In this case the bookmap now uses topicrefs with named keys and its keyrefs call the topic referenced in the mapref

```
<part keys="Reference-Material" keyref="reference_material">
    <topicref keys="Parts" keyref="parts"/>
    <topicref keys="Error-Messages" keyref="error_messages"/>
    <topicref keys="Error-Messages-Procedure" keyref="error_messages_procedure"/>
    <topicref keys="Model-II-Boot-Errors-Table" keyref="model_II_boot_errors_table"/>
    <topicref keys="Random-Tic-Tac-Toe" keyref="random_tic-tac-toe"/>
    <topicref keys="Random-Tic-Tac-Toe-Code" keyref="random_tic-tac-toe_code"/>
</part>
```

- From an Information Architect standpoint, you could start a new document that uses at least some of the topics from this document, use a mapref to reference them, and then use a topicref + keyref to call them in the new document

Keys and Conditional Processing

- In this DITA map, conditional processing for product="TRS80" references different keys than for product="TRS90"

```
<topicgroup processing-role="resource-only">
    <keydef product="TRS80" keys="product_info" href="topics/product_info_TRS80.dita"/>
    <keydef product="TRS80" keys="image_warehouse" href="topics/image_warehouse_TRS80.dit
    <keydef product="TRS90" keys="product_info" href="topics/product_info_TRS90.dita"/>
    <keydef product="TRS90" keys="image_warehouse" href="topics/image_warehouse_TRS90.dit
```

- This a relatively straightforward way of referencing different topics/images/URLs for different products, platforms, audience, etc.

Using ditaval with Keys on Example Document

Capabilities and Advantages

A brief overview of the additional features that the expansion interface can add to your TRS-80.

The Interface allows you to add the following Radio Shack modules to your system:

1. Screen Printer (26-1151)
2. Line Printer (26-1150)
3. Mini-Disk System (26-1160/26-1161)
4. Cassette Recorder number 2 (14-841)

The Screen Printer and Line Printer allow you to obtain hard copy (printed) information generated by your TRS-80.

The TRS-80 Mini-Disk System is a small version of the floppy disk. It provides vast storage space and much quicker access time than tape. The number 1 disk contains about 80,000 bytes of free space for files. Each additional disk has 89,600 bytes of file space. The Disk System has its own set of commands that allow manipulation of files and expanded abilities in file use. The TRS-80 Mini-Disk System uses sequential or random access. The disks will allow use of several additional LEVEL II commands.

Note: Because of the presence of a Disk Controller in the Expansion Interface, the computer will try to input the additional commands.

When the Expansion Interface is connected to the computer, it assumes that a Mini-Disk is connected. To use the Expansion Interface without a Mini-Disk, press the BREAK key on the TRS-80 keyboard. This will override the Mini-Disk mode and allow normal LEVEL II operation.

The use of two cassettes allows a much more efficient and convenient manner of updating data stored on tape. For example, if you have payroll data stored on tape, the information can be read, one item at a time, from Cassette Recorder number 1, then changed or added to and written out on Cassette Recorder number 2. The example cited is a very simple application; however, very powerful routines can be constructed to allow input and output of data using two tapes simultaneously.

Note: "This unit is designed to be used with Level II only. Do not use with level I.

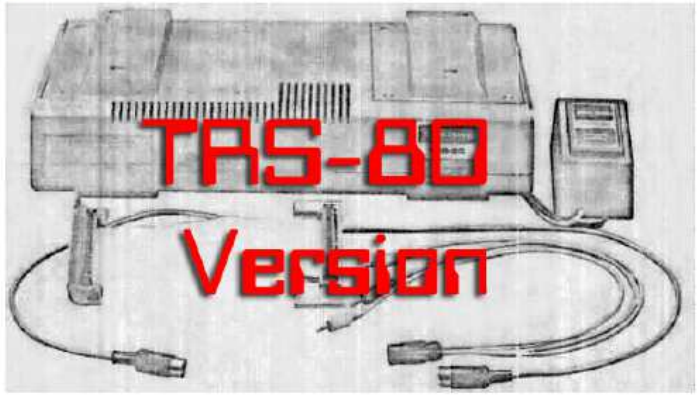


Figure 1: Expansion Interface.[x]

| Capabilities and Advantages | 5

Capabilities and Advantages

A brief overview of the additional features that the expansion interface can add to your TRS-90.

The Interface allows you to add the following Tandy modules to your system:

1. Screen Printer (26-1151)
2. Line Printer (26-1150)
3. Mini-Disk System (26-1160/26-1161)
4. Cassette Recorder number 2 (14-841)

The Screen Printer and Line Printer allow you to obtain hard copy (printed) information generated by your TRS-90.

The TRS-90 Mini-Disk System is a small version of the floppy disk. It provides vast storage space and much quicker access time than tape. The number 1 disk contains about 80,000 bytes of free space for files. Each additional disk has 89,600 bytes of file space. The Disk System has its own set of commands that allow manipulation of files and expanded abilities in file use. The TRS-90 Mini-Disk System uses sequential or random access. The disks will allow use of several additional LEVEL II commands.

Note: Because of the presence of a Disk Controller in the Expansion Interface Pro, the computer will try to input the additional commands.

When the Expansion Interface Pro is connected to the computer, it assumes that a Mini-Disk is connected. To use the Expansion Interface Pro without a Mini-Disk, press the BREAK key on the TRS-90 keyboard. This will override the Mini-Disk mode and allow normal LEVEL II operation.

The use of two cassettes allows a much more efficient and convenient manner of updating data stored on tape. For example, if you have payroll data stored on tape, the information can be read, one item at a time, from Cassette Recorder number 1, then changed or added to and written out on Cassette Recorder number 2. The example cited is a very simple application; however, very powerful routines can be constructed to allow input and output of data using two tapes simultaneously.

Note: "This unit is designed to be used with Level II only. Do not use with level I.

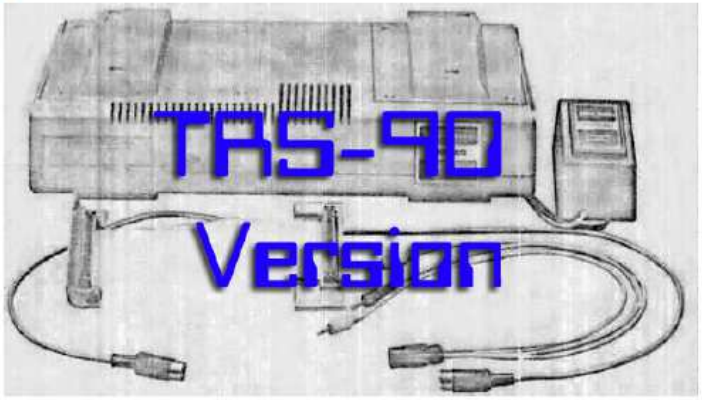


Figure 1: Expansion Interface Pro.[x]

Key Dependency Implications

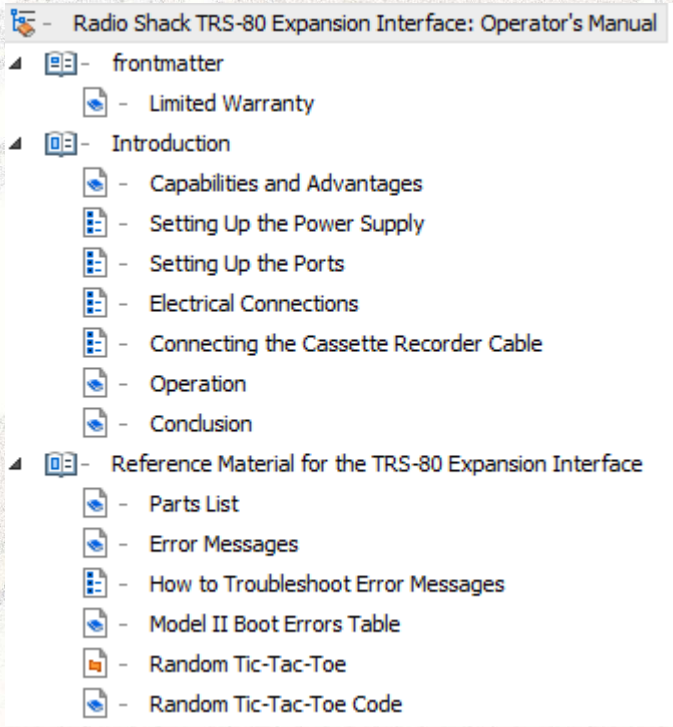
- Keys ultimately makes you DITA content map-dependent:
  - If you produce output from single topics and you want to use keys, you'll have to transition those publications to be map-based.
  - Writers will have to get used to working with topics in the context of a map if they want to see the full content resolved.

How Far Down the Rabbit Hole Do You Go?

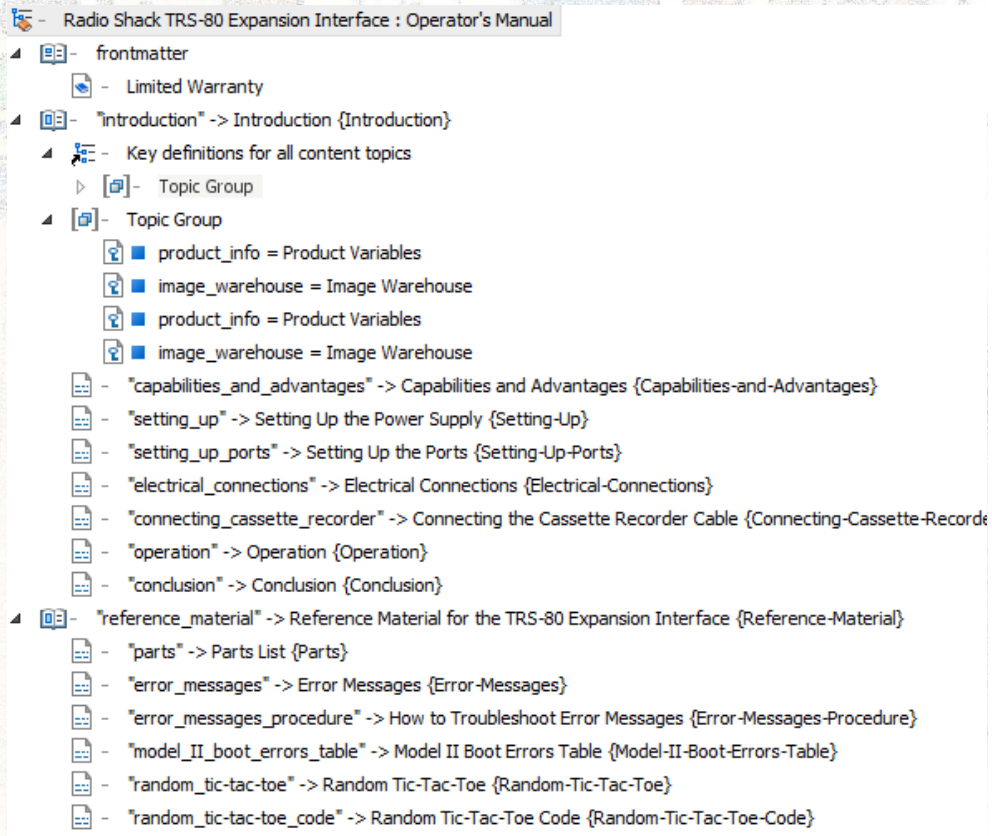
- You can go further than the examples seen here, but ultimately this adds to the complexity of using keys; documentation teams need to determine how far to abstract their keys

Trade off Between Flexibility and Opacity

- Ultimately your Information Architect needs to determine how deep to key, and strike a balance that the writers can easily work with



Original, un-keyed map in oXygen's Map View
Note that topic types are identified



Final example map. There is more to manage, and topic types are no longer identified

Some Wise Words on Key Complexity

“Some people would like to see keys used for all references. I think that is needlessly complex. References that are context-specific gain nothing from being keyed. It's only when you need to ‘flip-flop’ a term or a reference based on where you're using a topic that keys have value.

My advice is always to start as small as you can with keys. If you discover additional reuse potential down the road, you can always switch a direct reference to an indirect one.”

- Leigh White

Further Reading/Viewing

- Understanding DITA Keys and Key Spaces: www.oasis-open.org/committees/download.php/40946/understanding-dita-keys-and-key-spaces.pdf
- Thunderbird DITA Demo Content Collection: github.com/gnostyx/dita-demo-content-collection
- Scoped Keys in DITA 1.3 (Finally!): www.ixiasoft.com/download_file/force/26191/
- Get Keyed Up About DITA 1.3 Keys! www.youtube.com/watch?v=Ku6E9At0SE4
- Keys are the Key to Reuse [TC Dojo Open Session] www.youtube.com/watch?v=ea55hFTlkEI

Guide to the Code Examples

- All code examples can be found at: github.com/DITAWriter/dita_keys_examples
  - 01 - TRS80 Expansion Pack (No Keys) – Slides #15, #32
  - 01b - TRS80 Expansion Pack (Keys Values Defined in Map) – Slide #17
  - 02 - TRS80 Expansion Pack (Conkeyrefs for Product Names) – Slides #18-21
  - 03 - TRS80 Expansion Pack (Images Warehoused) – Slides #22-25
  - 04 - TRS80 Expansion Pack (Topicrefs Keyed) – Slides #26-27
  - 05 - TRS80 Expansion Pack (Conditionalized) – Slides #28-29, #32

- If you want a "Part 2" to this webinar, let Liz know!


Questions?
robertsk@ixiasoft.com
@KeithIXIASOFT
10in x 7in aspect 4/19/