



# Arm<sup>®</sup> Socrates<sup>™</sup>

Version 1.7.2

## User Guide

**Non-Confidential**

Copyright © 2018–2022 Arm Limited (or its affiliates).  
All rights reserved.

**Issue 04**

101399\_010702\_04\_en



# Arm® Socrates™

## User Guide

Copyright © 2018–2022 Arm Limited (or its affiliates). All rights reserved.

## Release Information

### Document history

Issue	Date	Confidentiality	Change
0100-00	25 July 2018	Confidential	First release for 1.0
0102-00	23 November 2018	Confidential	First release for 1.2
0103-00	8 February 2019	Confidential	First release for 1.3
010301-00	18 March 2019	Confidential	First release for 1.3.1
010302-00	24 May 2019	Confidential	First release for 1.3.2
010303-00	16 July 2019	Confidential	First release for 1.3.3
010304-00	1 October 2019	Confidential	First release for 1.3.4
0104-00	6 February 2020	Confidential	First release for 1.4
0105-00	25 September 2020	Non-Confidential	First release for 1.5
0105-01	23 October 2020	Non-Confidential	Second release for 1.5
1.6.0-01	18 June 2021	Non-Confidential	First release for 1.6.0. Document issue numbering style updated.
1.7.0-02	15 February 2022	Non-Confidential	First release for 1.7.0
1.7.1-03	17 June 2022	Non-Confidential	First release for 1.7.1 for Socrates REL
1.7.2-04	31 August 2022	Non-Confidential	First release for 1.7.2 for Socrates REL

## Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm. No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third party patents.

THIS DOCUMENT IS PROVIDED “AS IS”. ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow Arm’s trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>.

Copyright © 2018–2022 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349|version 21.0)

## Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

## Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on <https://support.developer.arm.com>.

To provide feedback on the document, fill the following survey: <https://developer.arm.com/documentation-feedback-survey>.

## Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

This document includes language that can be offensive. We will replace this language in a future issue of this document.

To report offensive language in this document, email [terms@arm.com](mailto:terms@arm.com).

# Contents

<b>1. Introduction.....</b>	<b>7</b>
1.1 Product revision status.....	7
1.2 Intended audience.....	7
1.3 Conventions.....	7
1.4 Useful resources.....	8
1.5 Feedback.....	9
<b>2. The Socrates IP Tooling platform.....</b>	<b>10</b>
2.1 About Socrates.....	10
2.2 Socrates at a glance.....	11
<b>3. Setup and maintenance tasks.....</b>	<b>14</b>
3.1 Workspaces.....	14
3.2 Product updates.....	14
3.3 Licensed IP locations.....	15
<b>4. Selecting, configuring, and building IP.....</b>	<b>17</b>
4.1 Finding IP.....	17
4.2 Associating licensed Arm IP.....	18
4.3 Managing projects.....	18
4.3.1 Project Explorer.....	18
4.3.2 Collaborative working.....	20
4.4 Configuring IP.....	22
4.5 Building IP.....	23
4.6 Adding interconnect IP with Interconnect Assistant.....	23
4.6.1 Component Generators.....	23
4.6.2 Create an empty System Specification.....	24
4.6.3 Add non-interconnect IP to the System Specification.....	25
4.6.4 Select and connect your IP with Interconnect Assistant.....	25
<b>5. Scripting API.....</b>	<b>28</b>
5.1 API scripts.....	28
5.2 Configured Component API.....	28

5.2.1 Cortex-A35 sample configure and build script.....	28
5.3 Extra utility API.....	29
<b>A. Command Line Interface.....</b>	<b>30</b>
A.1 Basic commands for Socrates CLI.....	30
<b>B. Document revisions.....</b>	<b>33</b>
B.1 Revisions.....	33

# 1. Introduction

## 1.1 Product revision status

The  $r_xp_y$  identifier indicates the revision status of the product described in this manual, for example,  $r1p2$ , where:

<b><math>r_x</math></b>	Identifies the major revision of the product, for example, $r1$ .
<b><math>p_y</math></b>	Identifies the minor revision or modification status of the product, for example, $p2$ .

## 1.2 Intended audience

This guide is written for system designers, integrators, engineers, and programmers who are configuring or building Arm IP using the Arm Socrates™ platform.

## 1.3 Conventions

The following subsections describe conventions used in Arm documents.

### Glossary

The Arm® Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: [developer.arm.com/glossary](https://developer.arm.com/glossary).

Convention	Use
<i>italic</i>	Citations.
<b>bold</b>	Terms in descriptive lists, where appropriate.
monospace	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
monospace <u>underline</u>	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments.  For example:  <pre>MRC p15, 0, &lt;Rd&gt;, &lt;CRn&gt;, &lt;CRm&gt;, &lt;Opcode_2&gt;</pre>

Convention	Use
<b>SMALL CAPITALS</b>	Terms that have specific technical meanings as defined in the <i>Arm® Glossary</i> . For example, <b>IMPLEMENTATION DEFINED</b> , <b>IMPLEMENTATION SPECIFIC</b> , <b>UNKNOWN</b> , and <b>UNPREDICTABLE</b> .



Recommendations. Not following these recommendations might lead to system failure or damage.



Requirements for the system. Not following these requirements might result in system failure or damage.



Requirements for the system. Not following these requirements will result in system failure or damage.



An important piece of information that needs your attention.



A useful tip that might make it easier, better or faster to perform a task.



A reminder of something important that relates to the information you are reading.

## 1.4 Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Access to Arm documents depends on their confidentiality:

- Non-Confidential documents are available at [developer.arm.com/documentation](https://developer.arm.com/documentation). Each document link in the following tables goes to the online version of the document.



- Confidential documents are available to licensees only through the product package.

**Table 1-2: Arm Publications**

Document name	Document ID	Licensee only
Arm® Socrates™ Installation Guide	101400	No

**Note**

Arm tests its PDFs only in Adobe Acrobat and Acrobat Reader. Arm cannot guarantee the quality of its documents when used with any other PDF reader.

Adobe PDF reader products can be downloaded at <http://www.adobe.com>

## 1.5 Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on <https://support.developer.arm.com>.

To provide feedback on the document, fill the following survey: <https://developer.arm.com/documentation-feedback-survey>.

## 2. The Socrates IP Tooling platform

Socrates™ enables you to browse, configure, and build Arm IP, including Arm Interconnect IP. You can then integrate the generated RTL and IP-XACT into Arm-based systems. Read this chapter if you are new to Socrates and want a quick overview of the product.

### 2.1 About Socrates

Socrates™ IP Tooling platform is an environment for exploring, configuring, and building Arm IP ready for integration into a *System on Chip* (SoC).

The following examples show you some of the ways you can use Socrates™.

#### Designing a SoC

If you want to:

- Investigate Arm® IP features, properties, and configuration options
- Use this information to decide which IP to use in your system or subsystem design
- See which IP you already have licenses for
- Provide your procurement team with a detailed list of part numbers for the IP that you want to license

You can:

- Browse the **IP Catalog** and view information in **Socrates Help**
- Filter the **IP Catalog** by IP name or supported protocols
- Export information from the **Bill of Materials** about required and alternative part numbers



You can open Socrates without a license. This enables you to browse the **IP Catalog** and view the videos.

---

#### Managing IP packages

If you want to:

- Ensure that the **IP Catalog** stays up-to-date with the latest release information
- Make finding the location of downloaded IP easier
- Update the **IP Catalog** entries so they are correctly associated with the latest IP packages

You can:

- View notifications for product and **IP Catalog** updates

- Enter shared download locations for all licensed IP packages
- Enable automatic refreshing of associations

## Develop a system with Arm IP

If you want to:

- Understand IP configuration options, interfaces, and protocols
- Integrate IP into a system
- Build IP-XACT and Verilog output
- Automate the configuration and build procedures

You can:

- Browse the **IP Catalog** and view technical documentation in **Socrates Help**
- Organize and configure IP in the **Project Explorer**
- See validation and configuration errors in the **Create Configured IP** window
- Automatically build simple IP as part of the configuration procedure
- Use the *Command Line Interface* (CLI), and script-based API, for script-based solutions

## Creating a high-level view of a SoC design

If you want to:

- Create a high-level view of your SoC design, containing configured IP instances and their high-level connections
- Choose the most suitable interconnect IP for your design, based on the high-level connectivity requirements of the other IP in your design
- Use this high-level view for further refinement and development of your design

You can:

- Use the **System Specification**, from the **Generators** group in the **IP Catalog**, to create an empty System Specification
- Add instances of your required non-interconnect IP to the System Specification
- Use **Interconnect Assistant** to select, configure, and add the most suitable interconnect IP to the System Specification, automatically creating all the required high-level connections

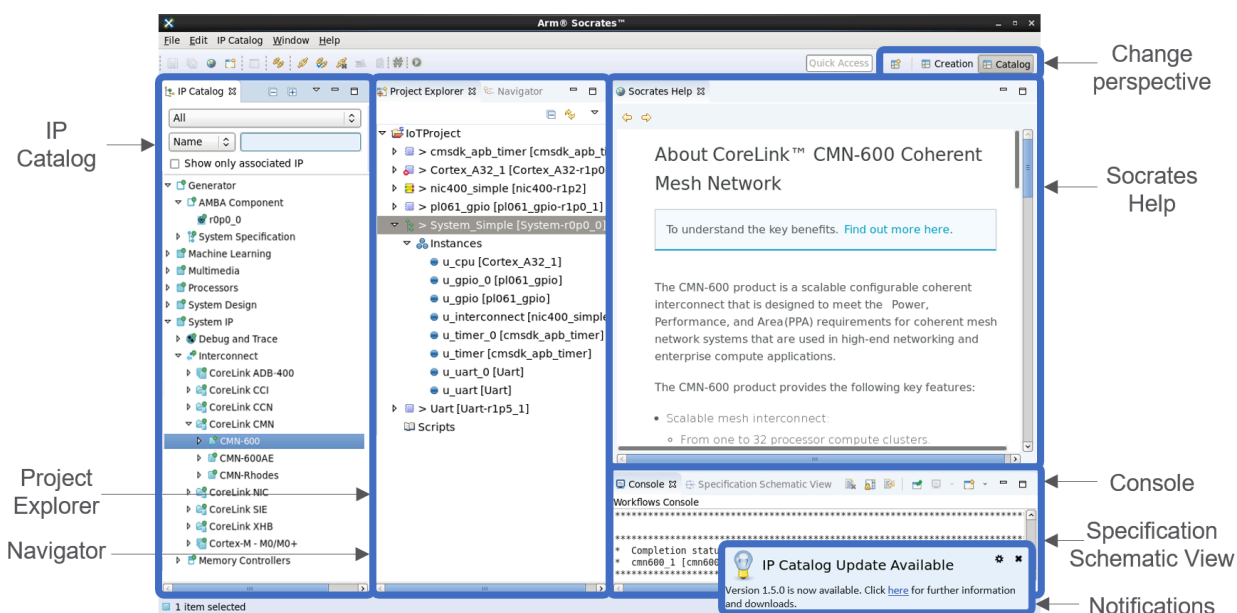
## 2.2 Socrates at a glance

Socrates™ has several panels, known as views, arranged into two perspectives.

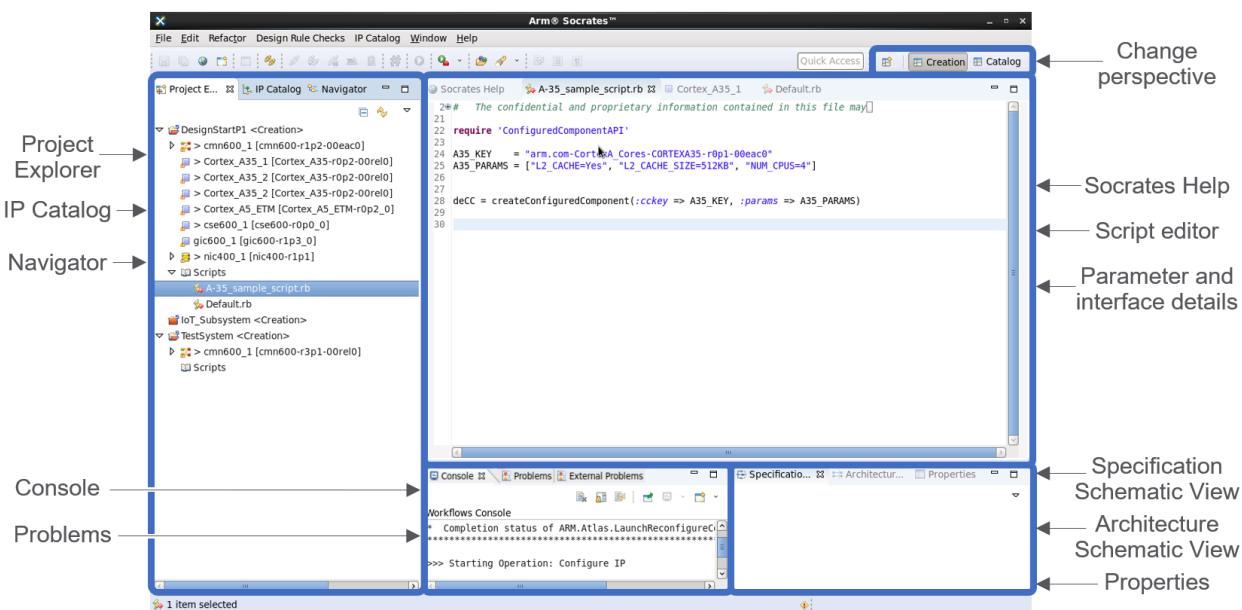
You can rearrange the panels to suit the way you work. To reset the default layouts, select **Window > Perspective > Reset Perspective...**

The following figures show the default layout of views in Socrates, after you close the Welcome page.

**Figure 2-1: Catalog perspective layout**



**Figure 2-2: Creation perspective layout**





Socrates is built on Eclipse, and not all of the Eclipse functionality is used or supported by Socrates. Therefore, if the functionality is not described in the Socrates documentation it might have unexpected behavior. For example, Socrates does not support the **Add project to working sets** option in the **File > Import > General > Existing Projects into Workspace** window.

---

## 3. Setup and maintenance tasks

To ensure that you have the latest version of the product and IP Catalog, you must perform several setup tasks when you first run Socrates™. Repeat these tasks periodically to stay up to date with the latest releases and functionality. Read this chapter about keeping Socrates™ up to date.



For details about installing Socrates, see *Arm® Socrates™ Installation Guide*.

### 3.1 Workspaces

Socrates™ stores project references in a workspace. When Socrates™ starts, you are asked to confirm the location that you want to use as a workspace. You can have multiple workspaces.

#### Procedure

To select a workspace:


1. Open the **Select a directory as workspace** window. Either:
  - When Socrates™ starts, the window opens automatically
  - When Socrates™ is running, select **File > Switch Workspace**
2. In the **Select a directory as workspace** window, either:
  - **Browse...** to the folder you want to use
  - Expand **Recent workspaces**, and select a previously used folder.
3. Optionally, to use the same workspace every time Socrates™ starts, select **Use this as default and do not ask again**.

### 3.2 Product updates

To ensure that you have the latest version of the product and the **IP Catalog**, you can install any available updates.

Every time Socrates™ starts, it automatically checks for the latest versions, and notifies you when they are available. You can view the available updates, start installing, and change the automatic setting in the **Updates** window. **IP Catalog** versions follow x.x.x numbering where "x.x.x" is a 3 or 4-digit version number, for example 1.6.0.

To open the **Updates** window, either:

- Select **IP Catalog > Update IP Catalog**.
- In the **IP Catalog**, click **Preferences**  and select **Updates**.

- Select **Window > Preferences**, expand **IP Catalog**, and select **Updates**.
- Click the download link in the notification message.

The following table describes the functionality of the **Updates** window.

**Table 3-1: Updates window functionality**

Name	Description
Automatically check for product updates	Change the setting to disable automatic checking for updates and notification. Default is selected.
Check for updates	Use to manually check for updates when the automatic checking is turned off
Updates section	Shows any available updates to the <b>IP Catalog</b> for current and newer versions of Socrates™
Install IP Catalog Updates	Launches the updated <b>IP Catalog</b> installation, follow the on-screen instruction



Socrates™ uses the `ARM_SYSOC_UPDATE_NOTIFICATIONS` environment variable to control whether the product update functionality is enabled. When this variable is set to `DISABLE`, all notifications and updates are disabled.

### 3.3 Licensed IP locations

Depending on your **Socrates** license, you can configure most IP without having a license for it. However, to build IP-XACT and Verilog output you must download a licensed IP product package to your system and associate it with the corresponding entry in the **IP Catalog**.


To download your licensed IP packages:

1. If you are a licensee, you can download the product package for that IP from Arm's Product Download Hub: <https://developer.arm.com/downloads>.
2. Install each package according to the instructions in the specific IP documentation.

See [4.2 Associating licensed Arm IP](#) on page 18 for information about associating your product.

To keep associations up-to-date, you can turn on automatic association in the **IP Library Path** window.

To open the **IP Library Path** window, either:

- Select **IP Catalog > IP Library Path Settings**.
- In the **IP Catalog**, click **Preferences** .
- Select **Window > Preferences**, expand **IP Catalog**, and select **IP Library Path**.

The following table describes the controls in the **IP Library Path** window that enable you to change the automatic association settings.

**Table 3-2: IP Library Path window functionality**

Name	Description
<b>Refresh associations on startup</b>	Changes the setting to automatically associate <b>IP Catalog</b> entries with IP packages that have been downloaded into the locations in <code>ARM_IP_LIBRARY_PATH</code> . Selected by default.
<b>Overwrite existing associations on startup</b>	When Socrates™ starts, this checkbox changes the setting to overwrite existing associations with the latest IP packages in the download locations
<b>ARM_IP_LIBRARY_PATH</b>	A colon (:) separated list of the download locations that you use for licensed Arm® IP
<b>Overwrite existing associations</b>	When an association is created manually, this checkbox changes the setting to overwrite existing associations with more recent IP packages in the download locations
<b>Refresh</b>	Check all associations for errors, create new associations, and overwrite existing associations with more recent IP packages in the download locations

### Related information

[4.2 Associating licensed Arm IP](#) on page 18



## 4. Selecting, configuring, and building IP

Socrates™ provides several types of technical information for each supported Arm IP, and a standardized way to configure and build IP. You organize your workspace into projects, ensure that the relevant IP has been associated with a downloaded IP package, configure, and build. Read this chapter if you are new to Socrates™ and want to understand how the product works.

### 4.1 Finding IP






Use the **IP Catalog** to find the IP you need, and to access relevant information about that IP. You can filter the **IP Catalog** by group, name, protocol, or valid associations.

If the **IP Catalog** is not visible, select **Window > Show View**, then select **IP Catalog**.

In the **IP Catalog**, you might see notification icons next to IP instances.

The following table describes the notification icons in the **IP Catalog**.

**Table 4-1: IP Catalog notification icons**

Name	Description
	The IP instance has not been associated with a licensed IP package
	You can build the IP instance, but some configurations require additional IP feature packages to build
	There is an error with an existing association
	You can configure and build this IP instance with the license you are using
	You cannot configure or build this IP instance with the license you are using

To show configuration guidance about each IP entry in the **IP Catalog**, select an IP entry. You can access further functionality using the **IP Catalog** menu.

To open the **IP Catalog** menu, right-click an entry in the **IP Catalog**.

The following table describes the options on the **IP Catalog** menu.

**Table 4-2: IP Catalog menu**

Name	Description
<b>Configure</b>	Opens the <b>Create Configured IP</b> window, enabling you to reconfigure the selected IP
<b>Associate IP Bundle</b>	Enables you to manually select a location of one or more licensed IP packages, and associate with the selected IP entries in the <b>IP Catalog</b>
<b>Check Associations</b>	Checks the current association for errors. Any errors that are found are highlighted.
<b>Remove Associations</b>	Removes the association of an IP entry with a licensed IP package

Name	Description
Bill of Materials	Shows the <b>Bill of Materials</b> for the selected IP, that lists the part numbers for required IP, alternative versions, features, and dependencies
Documentation	Shows the product documentation, if the entry has a valid association and documentation is included in the IP package
Properties	Opens information about the selected entry, for example links to further information on <a href="http://www.arm.com">www.arm.com</a> and <a href="http://developer.arm.com">developer.arm.com</a>

## 4.2 Associating licensed Arm IP

Depending on your **Socrates** license, you can configure most IP without having a license for it. However, to build IP-XACT and Verilog output, you must associate the **IP Catalog** entry with a licensed IP package.

### About this task

By default, Socrates™ automatically associates **IP Catalog** entries with IP packages that have been downloaded to your system, as long as the download locations are listed in `ARM_IP_LIBRARY_PATH`. See [3.3 Licensed IP locations](#) on page 15.

### Procedure

To manually associate an **IP Catalog** entry with a licensed IP package:

1. In the **IP Catalog**, select the folder or IP entry you want to associate with an IP package. If you select a folder, all entries within the folder are associated.
2. Right-click and select **Associate IP Bundle**.
3. Enter, or **Browse** to, the location of your downloaded package or packages. This location can be the same as, or different to, the locations in `ARM_IP_LIBRARY_PATH`.
4. If necessary, select **Overwrite existing bundles**.
5. Click **Finish**.

## 4.3 Managing projects

A project is a container for all the design data that is required to produce a SoC product.

Configured IP instances are saved to a project as part of the configuration procedure, so you must have a least one project created before you can configure your IP. See [4.4 Configuring IP](#) on page 22.

### 4.3.1 Project Explorer

Projects and configured IP are managed in the **Project Explorer** in the **Catalog** perspective.


If the **Project Explorer** is not visible:

- Select **Window > Show View** , and select **Project Explorer**.

In the **Project Explorer**, you can see notification icons next to any IP instances that have not been built, or associated with a downloaded IP package.

The following table describes the notification icons in the **Project Explorer**.

**Table 4-3: Project Explorer notification icons**

Icon	Description
> (greater than symbol)	The IP instance has not been built
 (exclamation symbol)	The IP instance has not been associated with a licensed IP package

You can access more functionality using the **Project Explorer** menu.

To open the **Project Explorer** menu:

- Right-click the **Project Explorer**, a project, or a configured IP instance, in the **Project Explorer**.



When your configuration is selected in the Project Explorer, the Schematic Viewer shows only the AMBA Designer interfaces. If you want to see all the interfaces, then you must disable the **Show AMBA interfaces only** option.

The following table describes the options for projects on the **Project Explorer** menu.

**Table 4-4: Project Explorer menu for projects**

Name	Description
<b>New Project...</b>	Opens the <b>New Project</b> window, follow the on-screen instructions to create a project.  You cannot import a project with the same name as an existing project. An extension such as _1 is added to the name, or you can customize the extension.
<b>Bill of Materials</b>	Shows the <b>Bill of Materials</b> for the selected project, that lists the part numbers for required IP, alternative versions, features, and dependencies
<b>Delete</b>	Opens the <b>Delete Project</b> window, where you can delete a project, and optionally, all the project data from the file system. Deleting data cannot be undone.
<b>Refresh</b>	Forces a refresh of the <b>Project Explorer</b> information
<b>Properties</b>	Opens information about the selected project. For example path variables, file locations, references to other projects, and details of any builders for the project
<b>Open Project</b>	Opens a project, where the underlying project data is saved in the workspace, but has not been loaded into the <b>Project Explorer</b> . Open projects are indicated with the open folder icon.
<b>Close Project</b>	Closes a project, removes the underlying project data from the <b>Project Explorer</b> . Closed projects are indicated with the closed folder icon.



You can import projects that have been created in different workspaces, or by different people. You can use this function to work collaboratively on projects. See [4.3.2 Collaborative working](#) on page 20.

The following table describes the options for configured IP instances on the **Project Explorer** menu.

**Table 4-5: Project Explorer menu for IP instances**

Name	Description
<b>New Project...</b>	Opens the <b>New Project</b> window, follow the on-screen instructions to create a project
<b>Interconnect Assistant</b>	Opens the <b>Interconnect Assistant</b> , enabling you to select, configure, and add an interconnect IP to the System Description
<b>Open</b>	Opens parameter and interface details for the selected IP
<b>Reconfigure</b>	Opens the <b>Create Configured IP</b> window, enabling you to reconfigure the selected IP
<b>Build</b>	Starts the rendering of IP-XACT and Verilog output for the selected IP. Usually, the build is performed as an automatic part of the configuration procedure. However, some IP require this separate action. Where this is the case, it is described in the specific <i>User Guide</i> for that IP.
<b>Bill of Materials</b>	Shows the <b>Bill of Materials</b> for the selected IP, that lists the part numbers for required IP, alternative versions, features, and dependencies
<b>Delete</b>	Deletes the selected IP, from the project and the file system
<b>Refresh</b>	Forces a refresh of the <b>Project Explorer</b> information
<b>Properties</b>	Opens file information for the selected IP, for example, path, filetype, location of file in the workspace, and permissions

## 4.3.2 Collaborative working

Socrates™ facilitates collaborative working by enabling you to reuse IP configurations.

### Sharing projects

You can use **File > Import** to import a project, and the contained IP, into your workspace. However, if you do not select **Copy projects into workspace**, you create a link to the original project. Any changes to one of the linked projects are reflected in the other.

Sharing a project enables more than one person to work on the project. For example, a shared project might work as follows:

1. You create a project, outside your workspace, in a shared location.
2. A colleague imports the project, ensuring **Copy projects into workspace** is not selected. They then add or reconfigure the IP.
3. You refresh the project and see the changed configurations.
4. Either of you can build or rebuild the IP.

You can use version control on the shared location, as long as the **logical** directory in the project is excluded.

## Sharing configured IP

You can export and import instances of configured IP between projects.

To export a configured IP instance `.soc` file:

1. In **Project Explorer**, right-click a configured IP instance, and select **Export**.
2. Expand **Socrates**, select **Export to Socrates Share (Collaboration)**, and click **Next**.
3. Select the configurations you want to export.
4. Browse to a location to save the configurations as `.soc` files.
5. Click **Finish**.

To import a configured IP instance:

1. In **Project Explorer**, right-click a project, and select **Import**.
2. Expand **Socrates**, select **IP configurations**, and click **Next**.
3. Browse to the location of a saved configuration. Confirm the project that you want to import the configurations into and click **Next**.
4. Select the configurations you want to import and click **Finish**.

## Sharing a configuration script

To export a configured IP to a script file:

1. In **Project Explorer**, right-click a configured IP instance, and select **Export**.
2. Expand **Socrates**, select **Export to Configure Script File**, and click **Next**.
3. Select the configurations you want to export.
4. Browse to a location to save the configurations as `.rb` files.
5. Click **Finish**.



A separate configuration file (`.ym1` or `.xm1` file) is generated for each configuration, and stored with the script file. If you copy a script file to a new location, you must also copy its associated configuration files to the same location.

---

To import a configuration script file:

1. In **Project Explorer**, right-click on **Scripts**.
2. Select **New Script...**
3. Enable **Link to file in file system** and browse for the file location. Make sure that the location contains both the script file and its associated configuration files.
4. Click **Finish**.

## Sharing configurations with Arm

To export a configured IP to send to Arm Support:

1. In **Project Explorer**, right-click a configured IP instance, and select **Export**.
2. Expand **Socrates**, select **Export to Arm Support**, and click **Next**.
3. Select the configurations you want to export.
4. Browse to a location to save the configurations as `.soc` files.
5. Click **Finish**.

## 4.4 Configuring IP

In the **Create Configured IP** window, you can configure most IP, add the configured IP to your project, and usually, build the IP.

### Before you begin

- You have created at least one project. See [4.3 Managing projects](#) on page 18.
- You have associated the IP with a licensed IP package. See [4.2 Associating licensed Arm IP](#) on page 18.

### About this task



You can use the Interconnect Assistant to help you choose and configure interconnect IP. See [4.6 Adding interconnect IP with Interconnect Assistant](#) on page 23.

To show configuration guidance in **Socrates Help**:

- In the **IP Catalog**, select an IP entry

To open the **Create Configured IP** window, either:

- Double-click an IP entry in the **IP Catalog**
- Right-click an IP entry in the **IP Catalog**, and select **Configure**

If you are prompted to open the **Creation** perspective, you must use a different configuration procedure. See the Socrates™ User Guide for the specific IP.

Most builds are an automatic part of the configuration procedure. However, if the IP entry has a synthesis step, you must complete the build manually. See [4.5 Building IP](#) on page 23.

### Procedure

To configure an IP instance:

1. In the **Create Configured IP** window, edit the parameters for your chosen IP.

2. If the selected IP has more than one page of parameters, complete each page, and click **Next**.
3. In **Project**, select the destination project for the configured IP.
4. In **Configured IP name suffix**, enter a name suffix or use the default.
5. Optionally, to override the build process, clear **Build Automatically**.
6. Click **Finish**. An instance of the IP with the specified parameter configuration is added to the project in the **Project Explorer**.

## Results

To reconfigure an IP instance, right-click an entry in the **Project Explorer**, then select **Reconfigure**.

## 4.5 Building IP

Building a configured IP renders IP-XACT and Verilog output into the project directory.

Some IP is built automatically after configuration. For complex IP, or if you want to run the build step manually from the editor, use the **Build** button.

## 4.6 Adding interconnect IP with Interconnect Assistant

Use the Interconnect Assistant to choose and add the most suitable interconnect IP to your SoC design.

### 4.6.1 Component Generators

Component generators can help you to design your SoC and configure your Arm IP. Component generators are listed under the **[Generator]** group in the **IP Catalog**.

#### System Specification Generator

Use the System Specification generator to create an empty System Specification and generate a high-level view of a SoC design. This view shows the configured IP instances in the design and the high-level connections between them, providing a starting point for further development. See [4.6.2 Create an empty System Specification](#) on page 24.

A System Specification:

- Contains only instances and high-level connections
- Is used for configuration only
- Cannot be associated with Arm IP
- Cannot be built

You can use System Specifications in multiple ways. Depending on the complexity of your SoC design, you may want to use multiple System Specifications, for example:

- To explore different design alternatives
- If different groups of IP in the design require different types of interconnect IP

## AMBA Component Generator

Use the AMBA Component generator to create a mock component that you can configure with AMBA interfaces and protocols. This component provides no other functionality.



The component can, for example, represent an instance of third-party IP in a System Specification. See [4.6.3 Add non-interconnect IP to the System Specification](#) on page 24.

---

An AMBA component:

- Contains only AMBA interfaces (requester and completer, AXI, ACE, AHB, CHI, APB)
- Is used for configuration only
- Cannot be associated with Arm IP
- Cannot be built

## 4.6.2 Create an empty System Specification

An empty System Specification is the starting point for your design.

### Before you begin

Ensure that you have a new or existing project. See [4.3 Managing projects](#) on page 18.

### Procedure

1. In the **Generators** group in the **IP Catalog**, double-click the **System Specification**.
2. In the **Create Configured IP** window:
  - a) Select your new project.
  - b) In **Configured IP name suffix**, enter a name suffix or use the default.
  - c) Click **Finish**.

### Results

An empty System Specification is added to the project.

### Next steps

[4.6.3 Add non-interconnect IP to the System Specification](#) on page 24.



### 4.6.3 Add non-interconnect IP to the System Specification

Create the instances of each non-interconnect IP required by your design.

#### Before you begin


Ensure that you have a project containing:

- A System Specification. See [4.6.2 Create an empty System Specification](#) on page 24.
- The non-interconnect IP that you want to use in your system. See [4. Selecting, configuring, and building IP](#) on page 17.



Use an AMBA component to represent third-party IP that you need. See [4.6.1 Component Generators](#) on page 23.

#### Procedure

1. In **Project Explorer**, select the System Specification.
2. In the **Instances** tab, click .
3. In the **Create Instances** window:
  - a) Select each **IP Configuration** that you want to use.
  - b) For each selected **IP Configuration**, enter the number of instances that you want to create.
  - c) Click **Next**.
4. In the **Instances** window, review and change instance names, or use the default values.
5. Click **Finish**, then save the project.

#### Results

The IP instances are listed in the System Specification. You can also see the IP instances and their interfaces in the **Architecture Schematic View**.

#### Next steps

[4.6.4 Select and connect your IP with Interconnect Assistant](#) on page 25.

### 4.6.4 Select and connect your IP with Interconnect Assistant

The **Interconnect Assistant** simplifies the task of choosing the most suitable interconnect IP to add to your System Specification, automatically creating all the required high-level connections.

#### Before you begin

Ensure that you have a project containing the instances of the non-interconnect IP that you want to use in your system. See [4.6.3 Add non-interconnect IP to the System Specification](#) on page 24.

#### About this task

First, the **Interconnect Assistant**:

- Determines the number and types of interfaces presented by all the IP instances in your system
- Ranks the available interconnect IPs in order of their support for the IP interfaces

You then choose:

- The interconnect IP to use
- The requester and completer interfaces to connect (see your product Technical Reference Manual for more information)

Finally, the **Interconnect Assistant**:

- Creates, configures, and instantiates the chosen interconnect IP.
- Creates the specified high-level connections between the interconnect IP instance and the other IP instances.

To open **Interconnect Assistant** for a System Specification, in **Project Explorer**, either:

- Select the System Specification, and click **Open Interconnect Assistant Wizard** on the toolbar.
- Right-click the System Specification, and choose **Interconnect Assistant**.

## Procedure

1. Open **Interconnect Assistant**.
2. In **Select Interconnect**, select the interconnect IP that you want to use.  
The window shows the total number of interfaces used by all the IP instances in the system that each interconnect IP supports. The number is shown separately for interfaces that are currently connected or not connected in the system. The interconnect IP are listed in descending order of the total number of interfaces that each can support.
3. In **Configure completer interfaces**, select each requester interface on an IP instance that you want to connect to a matching completer interface on the interconnect IP instance.  
The window shows:
  - All compatible interfaces present in the system, grouped by their parent IP instances
  - The interface types that are available in this interconnect IP, and any constraints that apply to each interface type



If your selections violate a constraint, an appropriate warning or error marker is shown. The Interconnect Assistant constraints are guidelines to follow in relation to interface selection. They do not consider all design constraints, nor guarantee valid design creation.

4. In **Configure requester interfaces**, repeat this process to select requester interface connections for the interconnect IP.
5. In **Create interconnect**, specify the configuration name suffix and instance name for the interconnect IP.
6. Click **Finish**.

## Results

You can see the connected IP in the **Architecture Schematic View**.



Based on its analysis of the configured IP in the System Specification, the **Interconnect Assistant** tries to provide sensible configuration values for each chosen interconnect IP. (If applicable, default memory maps, paths, and non-specified interfaces are created.) However, we cannot guarantee that any of these values are the most appropriate for your SoC design. You must review the generated configuration to ensure that it meets your specific requirements.

---

## 5. Scripting API


The script-based API enables tasks to be executed in script form, and output generated. You can use the **Project Explorer** or the *Command Line Interface* (CLI) to run scripts that you have created.

### 5.1 API scripts

Socrates IP Tooling platform supports scripts in Ruby, Tcl, and Python.

To create, edit, and run API scripts, you must be in the **Creation** perspective.

The following tasks can be accessed in **Project Explorer** by expanding the appropriate project and using the **Scripts** node:

- To create a script, right-click **Scripts** and select **New Script**.
- To open an existing script in the editor, double-click the script in the **Project Explorer**.
- To run a script, right-click the script in the **Project Explorer** and select **Run Script**.
- To run a script in Ruby, in the script editor, click **Run Script** , press **Ctrl+r**.

### 5.2 Configured Component API

The Configured Component API creates, configures, and builds configurable components.

To enable the Configured Component API in a script use:

```
require 'ConfiguredComponentAPI'
```

The API uses the **HLS** methods to create the IP Specification.



To use IP packages in the sample scripts, associate the appropriate IP packages.

#### 5.2.1 Cortex-A35 sample configure and build script

This sample script creates an instance of a Cortex®-A35 core, with the specified parameters, using the *Configured Component* API.

To open the sample script:

1. Open the **Creation** perspective.

2. In **Project Explorer**, expand your project and right-click on the **Scripts** node.
3. Select **New Script...**
4. In the dialog, select **Sample Cortex A35 Configure and Build**.

The **IP Catalog** *Configured Component* key is used to identify which component to create. Any unspecified parameters take on their default value.

1. Enable the *Configured Component* API:

```
require 'ConfiguredComponentAPI'
```

2. Define the *Configured Component* key from the IP Catalog, and the build parameters for the *Configured Component*. You might need to modify the supplied code to match the following parameters:

```
A35_KEY = "arm.com-CortexA_Cores-CORTEXA35-r0p1-00eac0"  
A35_PARAMS = ["L2_CACHE=Yes", "L2_CACHE_SIZE=512KB", "NUM_CPUS=4", "CRYPTO=No",  
              "NEON_FP=No"]
```

3. Create the *Configured Component* with the specified key and parameters:

```
deCC = createConfiguredComponent(:cckey => A35_KEY, :params => A35_PARAMS)
```

The *Configured Component* is automatically built and is located in the project folder in the **Project Explorer**. The other files that the build generates are located in the project folder in the workspace directory of the user.

The `createConfiguredComponent` method returns the *Configured Component* design element, which is stored as `decc`. Query this for more information on the built *Configured Component*.

For more information on the *Configured Component* API methods, see the Socrates Ruby API help in **Help > Help Contents > Arm Socrates Ruby API Manual > Ruby Configured Component API**.

## 5.3 Extra utility API

An extra utility API is available that supports a set of methods to create a Socrates script.

To enable the Ruby Utility API use:

```
require 'RubyUtilityAPI'
```

For more information, see the Ruby API help by selecting **Help > Help Contents > Arm Socrates Ruby API Manual**.

# Appendix A Command Line Interface

This appendix describes the *Command Line Interface* that Socrates™ provides to run workflows.

## A.1 Basic commands for Socrates CLI

You can use the CLI to run commands for much of the functionality in Socrates™.

Use the `--help` option to get more comprehensive help on the CLI:

```
socrates_cli --help
```



Note

For batch processing, the same composite workflow of CLI commands can be called several times during a single call using variable parameter values. The parameters that are used must be in a `<name>=<value>` pair, and the composite workflow being executed must understand them.

For example, `socrates_cli --project <project name> --flow <workflow name> parameter1=a --flow <workflow name> parameter1=b`.

Commands are run against the default workspace in `/home/<user>/armSocrates/workspace/`. To run commands against a different workspace, explicitly refer to it using the parameter `-data /<path to workspace>/`.



Caution

The `SOCRATES_USER_HOME` environment variable is used to set a new home area, before launching Socrates.

When running a large number of parallel CLI jobs, you can see Socrates home disk usage issues because all jobs are running using the same `~/armSocrates` home area. As a workaround, you can declare different Socrates home areas using an environment variable.

This type of error, which results from multiple parallel Socrates jobs sharing the resources of a single home area simultaneously, can manifest in different ways. Some example error messages are:

- An error occurred while locking file
- Error initializing storage

The following table describes some of the commands that can be used on the CLI.

**Table A-1: CLI commands**

Operation	Description	Command line usage
Select license option	Sets the license option: <ul style="list-style-type: none"> <li>socrates</li> <li>system_builder</li> <li>socrates_flexibleaccess_ms</li> <li>socrates_config_only</li> </ul>	socrates_cli --license <license option>
List catalog entries	Lists the current content of the IP Catalog. If <code>includeStatus</code> is true, then the association status of each entry is also printed. Use the optional argument <code>includeKeys</code> to return the <code>ConfigurableKey</code> used when creating a configured component.	socrates_cli --flow list.catalog.entries [includeStatus=true false] [includeKeys=true false]
Add new catalog entries	Adds new entries to the IP Catalog.  <code>newEntryArchive</code> refers to an archive file containing the new catalog entries.  <b>Note:</b> This action requires the installation to be writable in the installation environment.	socrates_cli --flow add.catalog.entries newEntryArchive=<path and file name to archive file from which new entries are extracted from>
Update the IP Catalog to the latest version	Installs all updates from the latest version of the IP Catalog, and updates the IP Catalog version number to the latest version.	socrates_cli --flow update.ip.catalog
Associate an IP bundle	Attempts to find catalog entry associations in the specified directory. If <code>overwrite</code> is true, then existing associations are overwritten.	socrates_cli --flow associate.bundle.ip bundleDirectory=<directory location of bundle to be associated> [overwrite=true false]
Create a project	Creates a project in the workspace.	socrates_cli --project <new project name> --flow AddNewProject
Create a configured component	Attempts to create a new configured component that is based on the configurable specified. The new configured component has its parameters set to the parameters that the <code>parameters</code> argument specifies. You must specify a suffix if the IP is configurable.	socrates_cli --project <existing project name> --flow create.configured.component configurableKey=<VLNV of the configurable component> [configuredComponentNameSuffix=<a name suffix for the configured component>] [parameters="parameter=value,parameter=value"] [subSystem="subsystem name"] [disableAutoBuild=true false]
Build a configured component	Executes the <code>BuildOperationReference</code> on the specified configured component.	socrates_cli --project <existing project name> --flow build.configured.component configuredComponentName=<a name for the configured component> [subSystem=<sub system name>]

Operation	Description	Command line usage
Export a configured component	Exports the specified configuration component to the specified file and directory.	<pre>socrates_cli --project &lt;project name&gt; --flow export.configuration -DoutputDir=&lt;output directory&gt; -DfileName=myExportedConfig -DconfiguredComponentName=&lt;name of the configuration to export&gt;</pre>
Import a configured component	Imports a previously exported configuration component.	<pre>socrates_cli --project &lt;project name&gt; --flow import.configuration -DexportedConfigFile=&lt;full pathname of exported configuration file including .soc suffix&gt; [-Doverwrite=&lt;true false&gt;] -DconfigFileToImport=&lt;name of the configuration to import&gt; [-Dsuffix=&lt;name suffix to add to the imported configuration&gt;] [-Dsubsystem=&lt;subsystem name&gt;]</pre>
Run a script	Executes the specified script as a scripted workflow.	<pre>socrates_cli --project &lt;existing project name&gt; --flow RunScript ScriptFile=&lt;location to script&gt;</pre>



# Appendix B Document revisions

This appendix describes the changes between released issues of this document.

## B.1 Revisions

This appendix describes changes between released issues of this book.

**Table B-1: Issue 0101-00**

Change	Location
First release	-

**Table B-2: Differences between issue 0101-00 and issue 0102-00**

Change	Location
Clarification of instructions to open build script	<a href="#">5.2.1 Cortex-A35 sample configure and build script</a> on page 28

**Table B-3: Differences between issue 0102-00 and issue 0103-00**

Change	Location
Added examples of the different ways to use Socrates™	<a href="#">2.1 About Socrates</a> on page 10
Added screenshots and descriptions of perspectives and views	<a href="#">2.2 Socrates at a glance</a> on page 11
Added tasks that are needed for setup and maintenance	<a href="#">3. Setup and maintenance tasks</a> on page 14
Added details about automatic updates for product and <b>IP Catalog</b>	<a href="#">3.2 Product updates</a> on page 14
Added a description of <b>IP Catalog</b> functionality, including new filtering, and association checking	<a href="#">4.1 Finding IP</a> on page 17
Added a description of <b>Project Explorer</b> functionality, including notification indicators	<a href="#">4.3.1 Project Explorer</a> on page 18
Added a description of <b>Bill of Materials</b> functionality	<a href="#">4.1 Finding IP</a> on page 17 and <a href="#">4.3.1 Project Explorer</a> on page 18
Added suggestions for collaborative working on projects	<a href="#">4.3.2 Collaborative working</a> on page 20

**Table B-4: Differences between issue 0103-00 and issue 010301-00**

Change	Location
Updated to 1.3.1 release	-

**Table B-5: Differences between issue 010301-00 and issue 010302-00**

Change	Location
Updated to 1.3.2 release	-

**Table B-6: Differences between issue 010302-00 and issue 010303-00**

Change	Location
Updated to include <i>Group</i> filter on <b>IP Catalog</b>	<a href="#">2.2 Socrates at a glance</a> on page 11, and <a href="#">4.1 Finding IP</a> on page 17.

**Table B-7: Differences between issue 010303-00 and issue 010304-00**

Change	Location
Updated to 1.3.4 release	-

**Table B-8: Differences between issue 010304-00 and issue 0104-00**

Change	Location
Added information about support for Eclipse functionality	<a href="#">2.2 Socrates at a glance</a> on page 11
Added alternative ways to open the <b>Updates</b> window	<a href="#">3.2 Product updates</a> on page 14
Added alternative ways to open the <b>IP Library Path</b> window	<a href="#">3.3 Licensed IP locations</a> on page 15
Added an icon to the <b>IP Catalog Notification icons</b> table regarding products where bundles need to be merged	<a href="#">4.1 Finding IP</a> on page 17
Added information about sharing a configuration script and sharing configurations with Arm	<a href="#">4.3.2 Collaborative working</a> on page 20

**Table B-9: Differences between issue 0104-00 and issue 0105-01**

Change	Location
First Non-Confidential release	-
Added an example of how to use Socrates to create a high-level view of a SoC design	<a href="#">2.1 About Socrates</a> on page 10
Added information about product usage analytics - what is collected, and how to disable collection	<a href="#">A.1 Basic commands for Socrates CLI</a> on page 30
Added how to start <b>Interconnect Assistant</b> from <b>Project Explorer</b>	<a href="#">Table 4-5: Project Explorer menu for IP instances</a> on page 20
Added how to use <b>Interconnect Assistant</b> to choose and add the most suitable interconnect IP a SoC design	<a href="#">4.6 Adding interconnect IP with Interconnect Assistant</a> on page 23
Added CLI command to update the IP Catalog to the latest version	<a href="#">A.1 Basic commands for Socrates CLI</a> on page 30

**Table B-10: Differences between issue 0105-01 and issue 1.6.0-01**

Change	Location
Editorial changes, including product version and document issue styles	Throughout
<ul style="list-style-type: none"> <li>Added <code>socrates_config_only</code> in the list of license options for the license flag</li> <li>Added information on analytics</li> <li>Clarified icon shown in Project Explorer</li> <li>Clarified required input for configuring component</li> </ul>	<a href="#">A.1 Basic commands for Socrates CLI</a> on page 30

**Table B-11: Differences between issue 1.6.0-01 and issue 1.7.0-02**

Change	Location
Inclusive language commitment statement updated	<a href="#">Inclusive language commitment</a> on page 4
Out-of-date data collection information removed	-
Removed content about analytics	<a href="#">2. The Socrates IP Tooling platform</a> on page 10, <a href="#">A.1 Basic commands for Socrates CLI</a> on page 30
New project information updated	<a href="#">4.3.1 Project Explorer</a> on page 18

Change	Location
Replaced "master" and "slave" terminology	<a href="#">4.6.1 Component Generators</a> on page 23, <a href="#">4.6.4 Select and connect your IP with Interconnect Assistant</a> on page 25
Reference to Technical Reference Manual added	<a href="#">4.6.4 Select and connect your IP with Interconnect Assistant</a> on page 25
--disable_analytics command removed, export and import information added, and note on disk usage issues	<a href="#">A.1 Basic commands for Socrates CLI</a> on page 30

**Table B-12: Differences between issue 1.7.0-02 and 1.7.1-03**

Change	Location
Updated document version for 1.7.1 REL release. No technical changes.	-

**Table B-13: Differences between issue 1.7.1-03 and 1.7.2-04**

Change	Location
Updated information and terminology for IP delivery	Throughout, including <a href="#">3.3 Licensed IP locations</a> on page 15