SoDA / File Agent Basics

- Module 1: Dashboard Overview
- Module 2: Dashboard Architecture
- Module 3: Dashboard Installation
- Module 4: Dashboard Operation and Customizations
- Module 5: SoDA / File Agent Basics
- Module 6: SoDA Agent Customizations
- Module 7: CSV Tool Basics and Operations
- Module 8: Dashboard API
- Module 9: Dashboard Administration

Learning Objectives

- **When you complete this module, you should be able to**
  - Understand the SoDA Agent architecture.
  - Understand the SoDA Agent command syntax
  - Understand the File agent command syntax

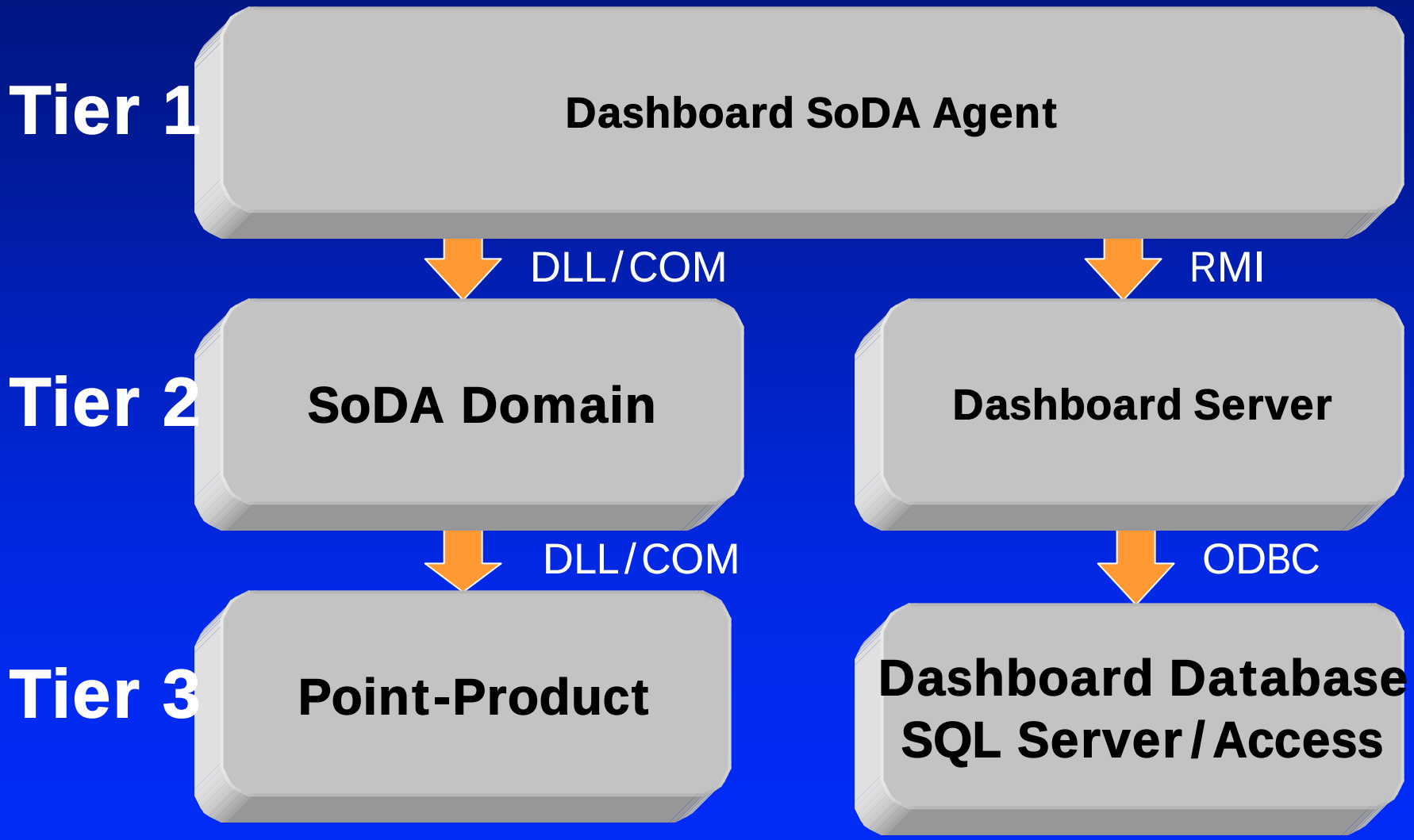
SoDA Agent Operations

- **SoDA Agent uses SoDA domains for collecting metrics**
- **Key functionality:**
  - Single API for collecting metrics from Rational tools
  - Agent supports all domains SoDA currently supports
    - Rose 98i
    - Requisite Pro
    - ClearQuest 1.x
    - Team Test 7.1
    - Microsoft Word
    - ClearCase Unix
    - Apex Unix
    - TestMate Unix

SoDA Agent Architecture

- **Windows NT Service**
  - Can be installed to automatically run when NT starts
- **Console application**
  - Interactive or manual collection
  - Debugging initialization file
- **Initialization file**
  - Collection instructions defined
  - Default Location: %windir%\SpcpSodaAgent.ini

SoDA Architecture (Windows NT)



SoDA Agent command

- **For debugging, run the console application:**
  - SpcpSodaAgent <ini file name>
  - Additional commands defined in initialization file

SoDA Agent Initialization Syntax

- **Format:**
  - "parameter=value"
  - Parameter names are not case-sensitive
  - Parameter values may be case-sensitive
  - Lines that do not start with a parameter name are ignored (treated as comment lines)
- **Global Parameters**
  - Parameters used to set up or initialize data collection
- **Domain Parameters**
  - Parameters used when collecting data using the SoDA domains
  - A single initialization file can contain multiple sets of Domain Parameters so that metrics can be collected from multiple tools using a single agent.

SoDA Agent Initialization File Definitions

Global Parameters

Parameter	Default	Case Sensitive	Optional	Description
Debug	False	No	Yes	Debug logging
RemoteHost	None	No	No	Host name of the database server
ServerRetries	120	No	Yes	If agent fails to connect to database server, it will retry *n* times
ServerTimeout	1	No	Yes	Time between retries
WhenToRun	Now	No	Yes	Controls the time the agent starts collecting. "Now" means immediately. Time format "hh:mm" between 00:00 and 23:59.

SoDA Agent Initialization File Definitions

Global Parameters

Parameter	Default	Case Sensitive	Optional	Description
RepeatInterval	0	No	No	Control how often (in hours) agent will run. If not specified, agent will run only once.
ShellCmd	None	No	Yes	Windows NT shell command to execute during initialization. Useful if you need to mount a network drive or execute ClearCase commands. Use multiple ShellCmd commands to execute multiple commands. All shell commands are executed each repeat interval.

SoDA agent Initialization File Definitions

Domain Parameters

Parameter	Default	Case Sensitive	Optional	Description
InputFile	None	No	Yes	Full path name of the data source (for non CQ domains) or the name of the CQ database.
Username	None	No	Yes	Username that has access to CQ database. If not specified, CQ will prompt for a username/password.
Password	None	No	Yes	Password for username used to access CQ database. If not specified, CQ will prompt for a username/password.

SoDA Agent Initialization File Definitions

Domain Parameters

Parameter	Default	Case Sensitive	Optional	Description
SodaDomain	None	No	Yes	Name of the SoDA domain required for parsing the input file.
ArtifactName	None	Yes	No	Artifact name to be assigned to the data collected. Also can be used to prefix dynamic artifact names
ArtifactSpec	None	Yes	No	Specification used to locate metric data within the InputFile.
HierarchyName	None	Yes	No	Name to assign to the Hierarchy created by this query
HierarchyRoot	None	Yes	No	Name to assign to the root node in the constructed hierarchy tree. Multiple levels in the hierarchy can be specified, separated by colons

SoDA Agent Initialization File Definitions

Domain Parameters

Parameter	Default	Case Sensitive	Optional	Description
HierarchySpec	None	Yes	Yes	Used to locate the hierarchy name for the metric item. Syntax must follow SoDA domain. Use the HierarchySpec in conjunction with the ParentHierarchySpec to recursively construct a hierarchy.
ParentHierarchy Spec	None	Yes	No	Used to build a hierarchy tree for the metric data item

SoDA Agent Initialization File Example

- **The following example initialization file is used to collect metrics from the Classics "demo" project that is delivered with Rational Suite**

Initialization File Global Parameters

```
[Configuration]
; Turn on Verbose debugging
debug=true
; Connect to the server on the local host
remoteHost=localhost
; Start running immediately
whenToRun=now
; Repeat every 24 hours (Note: the repeat interval is commented out so
; the agent will run only once)
;repeatInterval=24
; If the server is unavailable, retry 60 times, once every 5 seconds
serverTimeout=5
serverRetries=60
```


SoDA Agent Initialization File Example

Initialization File Domain Parameters

The domain parameters tell the SoDA agent where the source of the metric data is, how to extract the metric information from the data source, where in the database to store the data, and what artifact to associate with the data

```
; Specify the location of the input file
inputFile=C:\Program Files\Rational\Classics Demo\ClassicsRepo\Project\ClassicsPOS\Rose\classics.mdl
; Specify the SoDA domain to use.  In this case, we are processing a Rose model
sodaDomain=rose98

; Construct the static portion of the data hierarchy and artifact.  This initialization file will construct store
; the metrics in the “Classics” hierarchy under the Artifact name “UseCases”.  The Hierarchy will look like:
; Classics->Rose->...
hierarchyName=Classics
hierarchyRoot=Classics:Rose
artifactName=UseCases
```

SoDA Agent Initialization File Example

```
; Specify where in the model the UseCases are.  Note: AllUseCases is the SoDA query
      syntax
; to retrieve a list of every use case in a Rose model. Multiple SoDA keywords are
      separated
; by Colons.
artifactSpec=AllUseCases*

; For each use case that is found, dynamically create the hierarchy.  The leaf of the
; hierarchy will contain the name of the Parent Package
hierarchySpec=ParentPackage:Name

; Next, we want to dynamically construct the parent nodes recursively using the name
; of each package's parent package
parentHierarchySpec=ParentPackage#:Name
```

ArtifactSpec Meta Characters

The ArtifactSpec is used to determine where in the metric input file (e.g., Rose model, ClearQuest database, etc.) the metric primitive is located.

- **ArtifactSpec meta characters**
  - # indicates that subsequent HierarchySpec and ParentHierarchySpec queries should operate on this record

    ArtifactSpec=AllClasses*#:Properties*:Name=Instancing:Value=ArtifactName

    The # in this query indicates that the next HierarchySpec query should operate on a "Class" record, not a "Properties" record.

    AllClasses, Properties, Name, and Value are SoDA "domain" keywords.

    Instancing is the name of a Rose property

    ArtifactName is a SoDA "Agent" keyword used to dynamically build artifacts

  - * indicates that the query will return multiple values.

    AllClasses* returns a list of every Class in the Rose model

ParentHierarchySpec Meta Characters

- **ParentHierarchySpec meta characters**
  - # indicates that this is a recursive query - will repeat until no data is returned from SoDA
  - # also indicates that the next recursive query should start with this record

    ArtifactSpec=AllClasses*#:Properties*:Name=Instancing:Value=ArtifactName
    ParentHierarchySpec=ParentPackage#:Name

    - This ParentHierarchySpec first obtains a Class' ParentPackage name. Then, the name of the ParentPackage's ParentPackage is obtained. This repeats until we reach the top of the model. Note: Without the "#" in the ArtifactSpec, the first ParentPackage#:Name query would have started on a Class Property record, not the Class record.

  - An @ at the beginning of an entry indicates that the previous query should be stored, and that the next query should start at the root node. This is used on domains with "flat" hierarchies (ClearQuest)

    ArtifactSpec=Defect*#:State=ArtifactName
    ParentHierarchySpec=owner:login_name:@project:name

    - This ParentHierarchySpec obtains the defect's owner login_name, then obtains the defect's project name

SoDA Agent Summary

- **SoDA agent utilizes the SoDA domains to access point products**
- **SoDA can be run interactively (for debugging), or as an NT service**
- **SoDA agent is customized through an initialization file (%windir%\SpcpSodaAgent.ini)**
  - Global parameters include debug, remoteHost, retries, whenToRun, repeatInterval, ShellCmd
  - Domain parameters include inputFile, username, password, domain, artifactName, artifactSpec, hierarchyName, hierarchyRoot, hierarchySpec, parentHierarchySpec
- **Artifact and hierarchy specs follow SoDA domain syntax**

File Agent Operations

- **The File agent processes file system objects**
- **Key functionality:**
  - Can generate a SLOC count from a wildcard list of files
    - Very crude SLOC counter - not intended for detailed SLOC metric collection and analysis
  - Can generate file change metrics
    - Stores the date/time every time the File agent is run
    - Generates a count of files that been modified since the last run

File Agent command

- **The File agent and SoDA agent share a common Initialization file Parser:**
- **Command line syntax is indentical for the File and SoDA agents**
  - SpcpFileAgent <ini file name>
  - Additional commands defined in initialization file

File Agent Initialization File Syntax

- **Format:**
  - "parameter=value"
  - Parameter names are not case-sensitive
  - Parameter values may be case-sensitive
  - Lines that do not start with a parameter name are ignored (treated as comment lines)
- **Global Parameters**
  - Parameters used to set up or initialize data collection. The Global parameters are the same for the SoDA agent and File agent
- **Domain Parameters**
  - Parameters used when file metrics
  - A single initialization file can contain multiple sets of Domain Parameters so that metrics can be collected for multiple files using a single agent.
  - File name is always used as the hierarchy leaf node

File Agent Initialization File Example

Initialization File Domain Parameters

The domain parameters tell the File agent where the source of the metric data is, what type of metric information to collect (File Change or SLOC), where in the database to store the data, and what artifact to associate with the data

```
; Specify the location of the input file
inputFile= D:\spcp\SpcpSnapshot\SpcpSource\com\rational\spcp\*.java
; Specify that the File agent should count SLOCs
; Set SodaDomain=ModificationDate to generate file change metrics
SodaDomain=CountLines

; Construct the static portion of the data hierarchy and artifact.  This initialization file will
      construct store
; the metrics in the “Dashboard” hierarchy under the Artifact name “SLOC”.  The Hierarchy will
      look like:
; Dashboard->Utility->MyJavaUtility
hierarchyName=Dashboard
hierarchyRoot=Dashboard:Utility
artifactName=SLOC
```

File Agent Summary

- **File agent generated file metrics**
  - SLOC and File Change metrics
- **File can be run interactively (for debugging), or as an NT service**
- **FIle agent is customized through an initialization file (%windir%\SpcpFileAgent.ini)**
  - Same global parameters as the SoDA agent
  - Domain parameters include InputFile, domain, artifactName, hierarchyName, hierarchyRoot,
  - File name is always used as the hierarchy leaf node

Next